



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

“Creación de una red social mediante Google Maps”

Autor: Ken Yamaguchi Miguel

Tutor: Ángel García Crespo

Leganés, junio de 2015

Título: Creación de una red social mediante Google Maps
Autor: Ken Yamaguchi Miguel
Director: Ángel García Crespo

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Quisiera agradecer a la Universidad y a Ángel García Crespo por apoyarme y animarme a continuar con este proyecto y hacerlo al fin realidad, y por su paciencia y entrega que me han animado en momentos difíciles...

...GRACIAS a mis padres, que siempre han creído en mí y que con su educación y sus valores han hecho de mí la persona que soy hoy en día. Gracias a mis hermanos, por ser el reflejo en quién fijarme para hacer las cosas bien, y por despertar en mí la pasión por la informática...

... y GRACIAS a ti, Laura, por tu apoyo incondicional, tu conocimiento y por animarme tantas veces a seguir adelante, por ser mi compañera en este y en tantos proyectos que nos queda por vivir juntos.

Resumen

En este proyecto, titulado “Creación de una red social a través de Google Maps”, se procederá a documentar el proceso seguido para la creación de una aplicación web, integrada con la red social Facebook, y los servicios de geoposicionamiento y mapas que ofrece Google Maps.

La creación de la aplicación se ha realizado partiendo desde cero, y se han utilizado elementos de terceros para la integración con Facebook y el uso de Google Maps. En el momento del inicio de este proyecto, el concepto de red social basado en la localización era muy novedoso, y con el paso del tiempo, este tipo de servicios se han popularizado e integrado en diversas herramientas.

Palabras clave:

Las palabras clave que describen el proyecto son:

Aplicación web, red social, geoposicionamiento, facebook, google maps.

Abstract

In this project, entitled "Creating a social network through Google Maps", proceed to document the process for creating a web application, integrated with the social network Facebook, and services of geo and maps featuring Google Maps.

Creating the application has been made from scratch, and used items for integration with third-party Facebook and using Google Maps. At the time of initiation of this project, the concept of social network based on the location was very new, and with the passage of time, these services have become popular and integrated into various tools.

Keywords:

The keywords describing the project are:

Web application, social networking, geo, facebook, google maps

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	19
1.1 Introducción	19
1.2 Visión general	20
1.3 Motivación	21
1.4 Objetivos	22
1.5 Fases del desarrollo	23
1.6 Medios empleados.....	25
1.6.1 Sistemas Operativos.....	26
1.6.2 Herramientas para el desarrollo	27
1.6.3 Plataforma del producto	31
1.6.4 Lenguajes de programación.....	33
1.7 Estructura de la memoria	35
2. ESTADO DE LA CUESTIÓN Y ESTUDIO PREVIO	37
2.1 Estado de la cuestión	37
2.2 Estudio Previo	38
2.2.1 Redes Sociales.....	38
2.2.2 Servicios Basados en la Localización.....	39
2.2.3 Redes sociales con Servicios Basados en la Localización.....	42
2.2.4 Sistemas gestores de Bases de datos.....	43
2.2.5 Servidores Web	46
3. ANÁLISIS	50
3.1 Análisis del sistema.....	50
3.1.1 Descripción general del sistema.....	50
3.1.2 Requisitos de usuario.....	52
3.1.3 Casos de Uso.....	57
3.1.4 Requisitos software del sistema	70
4. DISEÑO E IMPLEMENTACIÓN	77
4.1 Diseño e implementación	77
4.1.1 Metodología de desarrollo.....	78
4.1.2 Arquitectura del sistema	79

ÍNDICE GENERAL

4.1.3 Modelo E/R.....	83
4.1.4 Modelo Relacional.....	84
4.1.5 Implementación	88
5. CONCLUSIONES	122
5.1 Conclusiones y líneas futuras	122
5.1.1 Conclusiones.....	123
5.1.2 Líneas futuras	124
6. PRESUPUESTO	126
6.1 Presupuesto del proyecto.....	126
6.1.1 Costes de personal.....	126
7. INSTALACIÓN	140
7.1 Instalación y configuración de una Aplicación en Facebook.....	140
7.1.1 Plataforma Facebook	140
7.1.2 Graph API Explorer	142
7.1.3 Instalar la aplicación Desarrollador.....	145
7.1.4 Configurar una nueva aplicación.....	145
8. DIAGRAMA GANTT.....	150
8.1 Diagrama Gantt	150

Índice de figuras

Ilustración 1 – Gráfico en el cual cada persona es representada por un punto (nodo) y la relación de amistad es representada con una línea (arista)	29
Ilustración 2 - Arquitectura típica de sistemas SBL.....	40
Ilustración 3 - Composición de sistemas Basados en la Localización	41
Ilustración 4 - Tabla Comparativa SGBD	45
Ilustración 5 - Tabla Comparativa Servidores Web	48
Ilustración 6 - Diagrama de Casos de Uso (Usuario registrado).....	57
Ilustración 7 - Diagrama de Casos de Uso (Usuario no registrado).....	57
Ilustración 8 - Diagrama de arquitectura Cliente-Servidor	79
Ilustración 9 - Diagrama de la arquitectura seleccionada	80
Ilustración 10 - Modelo típico de Aplicación Web	81
Ilustración 11 - Modelo de Aplicación Web de Facebook.....	82
Ilustración 12 - Diagrama Entidad-Relación.....	84
Ilustración 13 – Esquema del Modelo Relacional de la aplicación.....	85
Ilustración 14 - Incluir SDK de Facebook para PHP	89
Ilustración 15 - Instanciar objeto Facebook	89
Ilustración 16 - Script de comprobación de login en Facebook	90
Ilustración 17 - Código HTML para cargar la librería JavaScript del SDK de Facebook	90
Ilustración 18 - Código JavaScript que complementa el login en Facebook	91
Ilustración 19 - Diagrama de solicitud de token de acceso	92
Ilustración 20 - Ejemplo de llamada a la API de Facebook para PHP.....	92
Ilustración 21 - Código HTML para cargar la librería de JavaScript de Google Maps	93
Ilustración 22 - Código de ejemplo para cargar un mapa que muestra como centro la ciudad de Río de Janeiro	94
Ilustración 23 - Mapa de Google Maps con la ciudad de Río de Janeiro como centro.....	94
Ilustración 24 - Código de ejemplo para añadir un nuevo marcador	95
Ilustración 25 - Mapa de Río de Janeiro con un marcador.....	95
Ilustración 26 - Código de ejemplo para añadir contenido al marcador y un controlador de evento click que abra el contenido del marcador al hacer click sobre él.	97

ÍNDICE DE FIGURAS

Ilustración 27 - Al hacer click sobre el marcador se abre un globo con la información definida.....	97
Ilustración 28 - Código de ejemplo de búsqueda de direcciones	98
Ilustración 29 - Resultado de una búsqueda de una dirección y resultado en el mapa en forma de marcador	99
Ilustración 30 - Código de ejemplo para realizar una búsqueda inversa.....	100
Ilustración 31 - Resultado de una búsqueda por latitud-longitud (búsqueda inversa)	101
Ilustración 32 - Estructura de ficheros del proyecto	102
Ilustración 33 - Método conectar() de la clase CBaseDatos	103
Ilustración 34 - Método quote de la clase CBaseDatos.....	103
Ilustración 35 - Método obtenerFila() de la clase CBaseDatos.....	104
Ilustración 36 - Método obtenerFilas() de la clase CBaseDatos	104
Ilustración 37 - Método ejecutar() de la clase CBaseDatos	104
Ilustración 38 - Extracto de la definición de la clase CUsuarios	105
Ilustración 39 - Método crearUsuario() de la clase CUsuarios	105
Ilustración 40 - Método dameInfoUsuario() de la clase CUsuarios.....	106
Ilustración 41 - Método dameHistoricoUsuario() de la clase CUsuarios.....	106
Ilustración 42 - Método administrarPosiciones() de la clase CUsuarios	107
Ilustración 43 - Constantes globales definidas en el fichero config.php.....	108
Ilustración 44 - Código fuente del fichero FBInit.php	109
Ilustración 45 - Función que usa AJAX para carga asíncrona de datos	110
Ilustración 46 - Página de inicio de la aplicación.....	113
Ilustración 47 - Solicitud de acceso a Facebook para a través de la herramienta	114
Ilustración 48 - Permiso de acceso a la herramienta	114
Ilustración 49 - Permiso extendido de uso de la herramienta	115
Ilustración 50 - Mensaje de bienvenida y configuración de la herramienta.....	115
Ilustración 51 - Nueva posición en el mapa usando geoposicionamiento.....	116
Ilustración 52 - Publicación en Facebook de nueva posición en el mapa	116
Ilustración 53 - Resultado de buscar Nueva York en el mapa	117
Ilustración 54 - Vista con los amigos del usuario que usan la herramienta	117
Ilustración 55 - Nuevo comentario en la posición de un amigo	118
Ilustración 56 - Posición de un amigo con un comentario	118
Ilustración 57 - Mapa con las últimas posiciones almacenadas del usuario	119
Ilustración 58 - Configuración de la aplicación	120
Ilustración 59 - Solicitud de envío de invitaciones de la herramienta a amigos	120
Ilustración 60 - Modelo típico de Aplicación Web	141
Ilustración 61 - Modelo de Aplicación Web de Facebook.....	142
Ilustración 62 - Herramienta Graph API Explorer	143
Ilustración 63 - Obtener datos de amigos mediante Graph API.....	144
Ilustración 64 - Obtener datos de amigos mediante FQL.....	145
Ilustración 65 - Instalación de la aplicación Desarrollador.....	145
Ilustración 66 - Crear nueva aplicación.....	146
Ilustración 67 - Configuración de la página de lienzo	146
Ilustración 68 - Permisos necesarios para la aplicación	147
Ilustración 69 - Paso 1 autorización básica	148
Ilustración 70 - Paso 2 autorización extendida.....	148

Índice de tablas

Tabla 1 - Fases del desarrollo.....	24
Tabla 2 - Formato de tabla de especificación de requisitos	52
Tabla 3 - RU-01	52
Tabla 4 - RU-02	52
Tabla 5 - RU-03	52
Tabla 6 - RU-04	53
Tabla 7 - RU-05	53
Tabla 8 - RU-06	53
Tabla 9 - RU-07	53
Tabla 10 - RU-08	53
Tabla 11 - RU-09	53
Tabla 12 - RU-10	54
Tabla 13 - RU-11	54
Tabla 14 - RU-12	54
Tabla 15 - RU-13	54
Tabla 16 - RU-14	55
Tabla 17 - RU-15	55
Tabla 18 - RU-16	55
Tabla 19 – RU-17.....	55
Tabla 20 – RU-18.....	55
Tabla 21 - RU-19	56
Tabla 22 - RU-20	56
Tabla 23 - RU-21	56
Tabla 24 - RU-22	56
Tabla 25 - RU-23	56
Tabla 26 - RU-24	56
Tabla 27 - Formato de tabla de especificación de casos de uso.....	58
Tabla 28 – CU-01.....	58
Tabla 29 – CU-02.....	59

ÍNDICE DE TABLAS

Tabla 30 – CU-03	59
Tabla 31 – CU-04	60
Tabla 32 – CU-05	60
Tabla 33 – CU-06	61
Tabla 34 – CU-07	62
Tabla 35 – CU-08	62
Tabla 36 – CU-09	63
Tabla 37 – CU-10	63
Tabla 38 – CU-11	64
Tabla 39 – CU-12	64
Tabla 40 – CU-13	65
Tabla 41 – CU-14	66
Tabla 42 – CU-15	66
Tabla 43 – CU-16	67
Tabla 44 – CU-17	68
Tabla 45 – CU-18	69
Tabla 46 - RF-01	70
Tabla 47 - RF-02	70
Tabla 48 - RF-03	70
Tabla 49 - RF-04	71
Tabla 50 - RF-05	71
Tabla 51 - RF-06	71
Tabla 52 - RF-07	71
Tabla 53 - RF-08	72
Tabla 54 - RF-09	72
Tabla 55 - RF-10	72
Tabla 56 - RF-11	72
Tabla 57 - RF-12	73
Tabla 58 - RNF-01	74
Tabla 59 - RNF-02	74
Tabla 60 - RNF-03	74
Tabla 61 - RNF-04	74
Tabla 62 - RNF-05	75
Tabla 63 - RNF-06	75
Tabla 64 - Estimación por fases	127
Tabla 65 – Coste por fases	127

Capítulo 1

Introducción y objetivos

1.1 Introducción

El presente proyecto, “Creación de una red social mediante Google Maps”, tiene como objetivo desarrollar una aplicación web accesible a través de Facebook, basada en los servicios de geolocalización que proporciona Google Maps, que permita conectar a amigos en una red social. Se llevará a cabo el análisis, diseño e implementación de una interfaz web de usuario, en la que intervengan estos elementos.

1.2 Visión general

El objetivo principal de este proyecto consiste en la creación de una herramienta Web que proporcione al usuario la capacidad de interactuar con otros usuarios, convirtiéndose en una plataforma donde compartir información e intereses comunes.

La idea principal, parte de la utilización conjunta de la *geolocalización* o ubicación geográfica y las redes sociales, de tal manera, que un usuario pueda compartir con sus contactos su posición actual de manera gráfica, mediante el uso de Google Maps.

La herramienta utiliza las funciones proporcionadas por la red social Facebook para obtener la información de su perfil, y la lista de contactos del mismo. De esta manera, se intentará explotar el potencial social de Facebook, que proporciona una plataforma consolidada a través de la cual se puede llegar a un gran número de usuarios, de tal manera que el crecimiento de nuestra red pueda llegar a ser exponencial.

Para poder realizar el proyecto, previamente se ha realizado un estudio de las distintas herramientas que Facebook y Google Maps proporciona con el fin de realizar la integración con cualquier aplicación, tanto de escritorio, como Web.

Una vez estudiadas las posibilidades, se han integrado las herramientas de Google Maps y de Facebook para la creación de la aplicación

1.3 Motivación

Con este proyecto se pretende dar a conocer una visión general de posibles nuevas herramientas para situar geográficamente al usuario mediante sistemas basados en la localización.

Con el aumento de popularidad de las nuevas tecnologías que permiten al usuario estar conectado a la red constantemente, y con la facilidad e inmediatez de interactuar con otros mediante las redes sociales, el nacimiento de nuevas herramientas y la expansión de las existentes han dado lugar a una carrera por ser el primero en cubrir la cuota de mercado existente.

Herramientas como Foursquare, enfocada al uso con dispositivos móviles, permite compartir de manera sencilla tu ubicación geográfica a través de visitas a lugares de interés y negocios, de tal manera que tus contactos o los usuarios de la red social, pueden obtener información sobre los lugares visitados. Por otra parte, mediante un sistema de puntos y reconocimientos, el usuario puede beneficiarse de promociones en aquellos negocios a los que visita.

Otras redes sociales como Twitter, que permite añadir la ubicación de tus tweets, o Facebook, que tiene un funcionamiento similar a Foursquare, han añadido esta tecnología a sus herramientas.

El uso de herramientas que cubran la creciente necesidad por parte del usuario de compartir en cada momento, con el valor añadido de poder situar dicha información de manera gráfica en un mapa, han sido las motivaciones principales para la realización de este proyecto.

1.4 Objetivos

Con el fin de conocer en qué medida el proyecto ha sido exitoso o no, debemos definir los objetivos a cumplir a partir de este punto.

Como objetivo principal, se va a desarrollar una red social mediante la integración de Facebook y los mapas interactivos de Google Maps en un mismo sitio Web. Para cumplir con este objetivo deberemos conseguir los siguientes objetivos parciales:

Se deberá realizar un estudio para conocer en qué modo se puede integrar las dos plataformas, Facebook y Google Maps, en un mismo sistema y de manera externa, que sirva como herramienta única para la red social.

El sistema integrará el perfil del usuario de Facebook mediante las herramientas que éste proporciona, permitiendo de esta manera al usuario generar contenidos que se compartan en ambas plataformas.

El sistema debe ser capaz de generar mapas interactivos. Para ello, debe valerse de las herramientas proporcionadas por Google Maps a través de su API, cuyo funcionamiento será explicado más adelante.

No es objetivo por lo tanto realizar la implementación de un sistema de integración con las plataformas Facebook y Google Maps, sino utilizar las herramientas que estos proporcionan para realizar la misma.

El sistema será una aplicación Web, por lo tanto se debe tener especial cuidado en que sea lo suficientemente flexible para permitir la conexión con el mayor número de navegadores Web posible.

El sistema deberá ser compatible con el uso de navegadores avanzados que permitan la utilización de *geolocalización* a través de HTML5, tanto si el usuario utiliza esta tecnología en su navegador, como si no.

Se trata de generar contenido atractivo para el usuario, por lo que el sistema debe ser capaz de mostrar información sobre la actividad reciente e histórica sobre los amigos del usuario que tengan acceso a la red del sistema, es decir, aquellos que hayan instalado la aplicación.

Para finalizar, no es objetivo de este proyecto, por parte del sistema, la creación de perfiles de usuario. Para ello será necesario ser usuario de Facebook, plataforma que se encargará de todo ello.

Antes de llevar a cabo las tareas necesarias para desarrollar la aplicación, es necesario confeccionar un documento de análisis que incluya los siguientes puntos:

- Especificación de Requisitos (de usuario y funcionales)
- Diagrama de Casos de Uso, siguiendo el estándar UML.

1.5 Fases del desarrollo

A continuación se describen brevemente a las fases de que se compone el proyecto y sus fechas:

Nombre	Fecha de inicio	Fecha de fin	Recursos
Inicio del Proyecto	18/02/2009	18/02/2009	Ken Yamaguchi
Presentación Inicial	18/02/2009	12/03/2009	Ken Yamaguchi
Reunión explicativa	18/02/2009	18/02/2009	Ken Yamaguchi
Requisitos iniciales	19/02/2009	12/03/2009	Ken Yamaguchi
Análisis	13/03/2009	21/05/2009	Ken Yamaguchi
Estudio de Redes Sociales	13/03/2009	31/03/2009	Ken Yamaguchi
Estudio API Google Maps	30/03/2009	10/04/2009	Ken Yamaguchi
Estudio API Facebook	10/04/2009	28/04/2009	Ken Yamaguchi
Estudio Tecnología	04/05/2009	21/05/2009	Ken Yamaguchi
Diseño	01/06/2009	19/06/2009	Ken Yamaguchi
Arquitectura del sistema	01/06/2009	05/06/2009	Ken Yamaguchi
Diagrama E/R	12/06/2009	19/06/2009	Ken Yamaguchi
Implementación	31/08/2009	30/01/2013	Ken Yamaguchi
Estructura aplicación	31/08/2009	07/06/2010	Ken Yamaguchi
Creación de tablas	31/08/2009	04/09/2009	Ken Yamaguchi
Hoja de Estilos	07/09/2009	25/09/2009	Ken Yamaguchi
Implementación clases php	25/09/2009	07/06/2010	Ken Yamaguchi
Integración HTML5	10/11/2009	06/04/2010	Ken Yamaguchi
Integración JQuery	10/11/2009	20/11/2009	Ken Yamaguchi
Creación clases javascript	10/11/2009	06/04/2010	Ken Yamaguchi

GLOSARIO

Integración Google Maps	20/10/2009	09/02/2010	Ken Yamaguchi
Mostrar mapas	20/10/2009	21/10/2009	Ken Yamaguchi
Inserción cuadro de búsqueda	10/11/2009	26/11/2009	Ken Yamaguchi
Localización inversa	15/01/2010	02/02/2010	Ken Yamaguchi
Geolocalización	15/01/2010	09/02/2010	Ken Yamaguchi
Integración Facebook	21/10/2009	20/05/2010	Ken Yamaguchi
Mostrar información Facebook	21/10/2009	13/11/2009	Ken Yamaguchi
Mostrar Amigos	16/11/2009	04/12/2009	Ken Yamaguchi
Publicar en el muro	03/12/2009	30/12/2009	Ken Yamaguchi
Invitaciones amigos	10/05/2010	20/05/2010	Ken Yamaguchi
Configuración de la aplicación	20/05/2010	28/05/2010	Ken Yamaguchi
Actualización Graph API de Facebook	17/12/2012	30/01/2013	Ken Yamaguchi
Documentación del proyecto	16/12/2013	28/02/2014	Ken Yamaguchi
Documentación	16/12/2013	27/02/2014	Ken Yamaguchi
Presupuesto	28/02/2014	28/02/2014	Ken Yamaguchi
Fin del Proyecto	03/03/2014	03/03/2014	Ken Yamaguchi

Tabla 1 - Fases del desarrollo

1.6 Medios empleados

A continuación, se incluye un esquema resumen de los elementos que se han utilizado para el desarrollo de este proyecto. Las herramientas que aquí se muestran aparecen descritas con más detalle en los siguientes apartados.

- Sistemas operativos:
 - Microsoft Windows 7
- Herramientas para el desarrollo:
 - Editor de Documentación:
 - GantProject 2.6.4
 - Editor de UML:
 - Dia 0.97.2
 - Entorno de desarrollo:
 - NetBeans IDE 7.0
 - Planificación de proyectos:
 - MS Project 2007
 - Realización de capturas:
 - GreenShot 1.1.7.17
 - API Google:
 - API de Google Maps (JavaScript Maps API V3)
 - API Facebook:
 - Open Graph API
 - API AJAX:
 - JQuery
- Plataforma del producto:
 - Servidor web: Apache 2. Versión 2.2.12
 - PHP: Versión 5.2.10
 - PhpMyAdmin: Version 3.2.2
 - MySQL. Versión 5.1.37
- Lenguajes de programación:
 - HTML y CSS
 - JAVASCRIPT, AJAX
 - PHP
 - MySQL

A continuación se incluyen las descripciones completas acerca de las herramientas y tecnologías mencionadas.

1.6.1 Sistemas Operativos

1.6.1.1 Microsoft Windows 7 Starter

Windows 7 es una de las versiones más reciente de Microsoft Windows, línea de sistemas operativos producida por Microsoft Corporation. Esta versión está diseñada para uso en PC, incluyendo equipos de escritorio en hogares y oficinas, equipos portátiles, tablet PC, netbooks y equipos media center. El desarrollo de Windows 7 se completó el 22 de julio de 2009, siendo entonces confirmada su fecha de venta oficial para el 22 de octubre de 2009 junto a su equivalente para servidores Windows Server 2008 R2.

A diferencia del gran salto arquitectónico y de características que sufrió su antecesor Windows Vista con respecto a Windows XP, Windows 7 fue concebido como una actualización incremental y focalizada de Vista y su núcleo NT 6.0, lo que permitió mantener cierto grado de compatibilidad con aplicaciones y hardware en los que éste ya era compatible. Sin embargo, entre las metas de desarrollo para Windows 7 se dio importancia a mejorar su interfaz para volverla más accesible al usuario e incluir nuevas características que permitieran hacer tareas de una manera más fácil y rápida, al mismo tiempo que se realizarían esfuerzos para lograr un sistema más ligero, estable y rápido.

Diversas presentaciones ofrecidas por la compañía en 2008 se enfocaron en demostrar capacidades multitáctiles, una interfaz rediseñada junto con una nueva barra de tareas y un sistema de redes domésticas simplificado y fácil de usar denominado «Grupo en el hogar», además de importantes mejoras en el rendimiento general del sistema operativo.

[WINDOWS]

1.6.2 Herramientas para el desarrollo

1.6.2.1 MS Word 2007

Microsoft Word 2007 es un software destinado al procesamiento de textos, creado por Microsoft e integrado en la suite informática Microsoft Office.

Ayuda a crear documentos de aspecto profesional al proporcionar un conjunto completo de herramientas para crear y dar formato a los documentos en la nueva interfaz de usuario de Microsoft Office Fluent.

Permite revisar, comentar, y comparar los cambios realizados en documentos compartidos con otros usuarios.

[WORD]

1.6.2.2 Dia

Dia es una aplicación informática de propósito general para la creación de diagramas, desarrollada como parte del proyecto GNOME. Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

Dia está diseñado como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas. Actualmente se incluyen diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, diagramas de circuitos eléctricos, etc. Nuevas formas pueden ser fácilmente agregadas, dibujándolas con un subconjunto de SVG e incluyéndolas en un archivo XML.

El formato para leer y almacenar gráficos es XML (comprimido con gzip, para ahorrar espacio). Puede producir salida en los formatos EPS, SVG y PNG.

También conviene recordar que Dia, gracias al paquete dia2code, puede generar el esqueleto del código a escribir, si utilizáramos con tal fin un UML.

[DIA]

1.6.2.3 NetBeans IDE 7.0

NetBeans IDE es un entorno de desarrollo integrado galardonado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, empresariales, de escritorio y aplicaciones móviles utilizando la plataforma Java, así como PHP, JavaScript y Ajax, Groovy y Grails, y C / C++.

El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece una amplia documentación y recursos de capacitación, así como una variada selección de plugins de terceros.

NetBeans IDE 7.0 introduce soporte de idiomas para el desarrollo de la especificación de Java SE 7 con características del lenguaje JDK 7. La versión también ofrece una integración mejorada con el servidor Oracle WebLogic, así como soporte para Oracle Database y GlassFish 3.1. Otros puntos destacados incluyen Maven 3 y soporte de edición de HTML 5, un nuevo diseñador GridBagLayout para un mejor desarrollo de GUI; mejoras en el editor de Java, y más.

NetBeans IDE 7.0 está disponible en Inglés, Chino, portugués, japonés, ruso y brasileño simplificado.

[NETBEANS]

1.6.2.4 GanttProject 2.6.4

GanttProject es un gestor de proyectos basado en Java, bajo licencia GPL, que funciona tanto en Windows, Linux y Mac OS. Permite introducir datos reales de evolución y realizar un completo seguimiento de cada una de las partes de que se compone el proyecto. Permite gestionar y controlar tanto las tareas que componen un proyecto, como los recursos que se utilizan para su desarrollo. Posee varias opciones de exportar los proyectos gestionados como son formato MS Project, HTML, PDF e imagen.

[GANTTPROJECT]

1.6.2.5 GreenShot

GreenShot es una utilidad optimizada para la productividad. Permite realizar capturas de pantalla en segundos. Es capaz de realizar capturas de ventanas, regiones o de la pantalla completa. Soporta infinidad de formatos.

Se distribuye bajo una licencia GNU General Public License versión 2.0 (GPLv2).

[GREENSHOT]

1.6.2.6 Graph API de Facebook

La plataforma Facebook es un entorno de software que provee la red social Facebook para que los desarrolladores puedan crear aplicaciones con acceso a los datos en Facebook. La plataforma ofrece un conjunto de interfaces de programación y herramientas que permiten a los desarrolladores integrar funcionalidades de la red en aplicaciones de terceros, mediante "Open Graph API".

Open Graph API es el núcleo de la plataforma Facebook. Permite a los desarrolladores leer y escribir datos en Facebook. La API representa de manera simple y consistente una vista de la gráfica social de Facebook, que representa uniformemente objetos en el gráfico (por ejemplo personas, eventos y páginas) y las conexiones entre ellos (por ejemplo, las relaciones de amistad entre personas, el contenido compartido, y las etiquetas de las fotos).

El concepto "Open Graph" hace referencia en el contexto de Internet a un gráfico que muestra las relaciones personales de los usuarios de Internet, una red social, donde la palabra gráfico se ha tomado de la teoría de grafos para enfatizar que el análisis matemático riguroso, se aplicará como se oponen a la representación relacional en una red social. El gráfico social se ha referido como "el levantamiento de mapas mundiales de todo el mundo y cómo están relacionadas". [FACEBOOK].

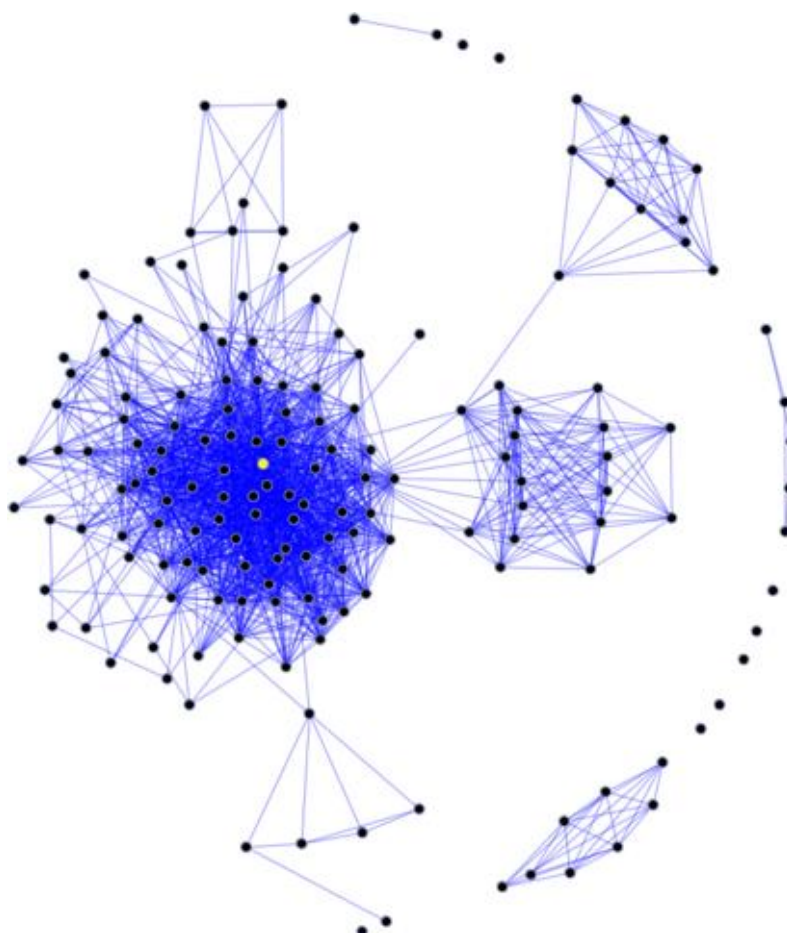


Ilustración 1 – Gráfico en el cual cada persona es representada por un punto (nodo) y la relación de amistad es representada con una línea (arista)

[FACEBOOK_DEVELOPERS]

1.6.2.7 API de Google Maps (JavaScript Maps API V3)

El API de JavaScript de Google Maps permite a los desarrolladores insertar Google Maps en una página web. La versión 3 de esta API está especialmente diseñada para proporcionar una mayor velocidad y que se pueda aplicar más fácilmente tanto a móviles como a las aplicaciones de navegador de escritorio tradicionales.

El API proporciona diversas utilidades para manipular mapas y para añadir contenido al mapa mediante diversos servicios, permitiéndote crear sólidas aplicaciones de mapas en sitios web.

Google Maps es un servicio de Google que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

Desde su lanzamiento en febrero de 2005, la aplicación cartográfica de Google ha conmocionado a la comunidad de desarrolladores. Si bien sus principios técnicos de base eran ya conocidos, incluso utilizados desde hacía tiempo, la aplicación de Google los combina de manera inteligente, y sobre todo ofrece una accesibilidad sin igual.

Varios aspectos de Google Maps son los responsables de su facilidad de uso por cualquier usuario: el sistema de deslizamiento de imagen, acoplado a la carga dinámica de nuevas imágenes; la adaptación del mapa al tamaño de ventana del navegador; la

GLOSARIO

interfaz minimalista; la posibilidad de cambiar de tipo de mapa en un clic y un largo etcétera.

Como todas las demás aplicaciones Google, Maps descansa poderosamente sobre la utilización de JavaScript. La carga y el deslizamiento de imagen no podrían efectuarse sin este código.

[GOOGLEMAPS]

1.6.3 Plataforma del producto

1.6.3.1 Apache2

El Proyecto Apache HTTP Server es un esfuerzo de desarrollo de software de colaboración cuyo objetivo es crear un servidor HTTP (web) que fuera robusto, con muchas características y cuyo código fuente fuera libre y estuviera a disposición de cualquiera. El proyecto está gestionado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, que usan Internet y la Web para comunicarse, planear y desarrollar el servidor y su documentación relacionada. Este proyecto forma parte de la Fundación de Software Apache. Además, cientos de usuarios han contribuido con ideas, código y la documentación para el proyecto.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

[APACHE]

1.6.3.2 phpMyAdmin

phpMyAdmin es una herramienta de software libre escrito en PHP, con la intención de manejar la administración de MySQL a través de la Web. phpMyAdmin es compatible con una amplia gama de operaciones en MySQL, MariaDB y Dizzle. Se pueden realizar operaciones de uso frecuente (gestión de bases de datos , tablas, columnas , relaciones , índices, usuarios , permisos, etc) mediante la interfaz de usuario, además de poder ejecutar directamente cualquier sentencia SQL .

phpMyAdmin tiene una amplia gama de documentación. Para facilitar el uso de una amplia gama de personas, phpMyAdmin está siendo traducido a 72 idiomas y es compatible con los dos idiomas LTR y RTL.

phpMyAdmin ha ganado varios premios. Entre otros, fue elegida como la mejor aplicación PHP en varios premios y ha ganado todos los años los Premios de la Comunidad SourceForge.net como "Mejor herramienta o utilidad para los administradores de sistemas".

[PHPMYADMIN]

1.6.3.3 MySQL

MySQL es un software de base de datos de código abierto más popular del mundo, con más de 100 millones de copias de su software descargado o distribuidos por toda su historia. Con su velocidad superior, confiabilidad y facilidad de uso, MySQL se ha convertido en la opción preferida para Web, Web 2.0, SaaS, ISV, empresas de telecomunicaciones y los administradores de TI empresarial con visión de futuro, ya que elimina los problemas principales asociados con el tiempo de inactividad, mantenimiento y administración para las aplicaciones modernas, en línea.

Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo utilizan MySQL para ahorrar tiempo y dinero alimentar sus sitios Web de alto volumen, los sistemas críticos de negocio y paquetes de software - incluyendo a los líderes de la

GLOSARIO

industria como Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, y Booking.com.

La oferta de MySQL estrella es MySQL Enterprise, un conjunto completo de software de producción probada, herramientas de monitorización proactivas, y servicios de apoyo de alta calidad disponibles en una suscripción anual asequible.

MySQL es una parte clave de LAMP (Linux, Apache, MySQL, PHP/Perl/Python), la pila de software empresarial de código abierto de rápido crecimiento. Cada vez más empresas están utilizando LAMP como una alternativa a las costosas pilas de software propietario debido a su menor costo y la libertad de la plataforma de lock-in.

MySQL fue originalmente fundada y desarrollada en Suecia por dos suecos y un finlandés: David Axmark, Allan Larsson, y Michael "Monty" Widenius, que habían trabajado juntos desde la década de 1980.

[MYSQL]

1.6.4 Lenguajes de programación

1.6.4.1 HTML

HTML (HyperText Markup Language, Lenguaje de Marcado de Hipertexto) hace referencia al lenguaje de marcado para la construcción de páginas web. Es un estándar que, en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, etc.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

[HTML]

1.6.4.2 CSS

CSS (Cascading Style Sheets, Hojas de Estilo en Cascada) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

[CSS]

1.6.4.3 PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término *PHP*.

[PHP]

1.6.4.4 SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

Los orígenes del SQL están ligados a los de las bases de datos relacionales. En 1970 E. F. Codd propone el modelo relacional y asociado a este un sublenguaje de acceso a los datos basado en el cálculo de predicados. Basándose en estas ideas, los laboratorios de IBM definen el lenguaje SEQUEL (Structured English Query Language) que más tarde sería ampliamente implementado por el sistema de gestión de bases de datos (SGBD) experimental System R, desarrollado en 1977 también por IBM. Sin embargo, fue Oracle quien lo introdujo por primera vez en 1979 en un programa comercial.

El SEQUEL terminaría siendo el predecesor de SQL, siendo este una versión evolucionada del primero. El SQL pasa a ser el lenguaje por excelencia de los diversos sistemas de gestión de bases de datos relacionales surgidos en los años siguientes y es por fin estandarizado en 1986 por el ANSI, dando lugar a la primera versión estándar de este lenguaje, el "SQL-86" o "SQL1". Al año siguiente este estándar es también adoptado por la ISO.

Sin embargo, este primer estándar no cubre todas las necesidades de los desarrolladores e incluye funcionalidades de definición de almacenamiento que se consideró suprimirlas. Así que, en 1992, se lanzó un nuevo estándar ampliado y revisado del SQL llamado "SQL-92" o "SQL2".

En la actualidad el SQL es el estándar de facto de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

El lenguaje SQL es un lenguaje declarativo. Es decir, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución.

El orden de ejecución interno de una sentencia puede afectar gravemente a la eficiencia del SGBD, por lo que se hace necesario que éste lleve a cabo una optimización antes de su ejecución. Muchas veces, el uso de índices acelera una instrucción de consulta, pero ralentiza la actualización de los datos. Dependiendo del uso de la aplicación, se priorizará el acceso indexado o una rápida actualización de la información. La optimización difiere sensiblemente en cada motor de base de datos y depende de muchos factores.

Existe una ampliación de SQL conocida como FSQL (Fuzzy SQL, SQL difuso) que permite el acceso a bases de datos difusas, usando la lógica difusa. Este lenguaje ha sido implementado a nivel experimental y está evolucionando rápidamente.

[SQL]

1.7 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo.

1. Capítulo 1:
En el primer capítulo, se engloba de manera resumida, el marco donde se encuadra el proyecto, y los objetivos que se persiguen.
2. Capítulo 2:
En el segundo capítulo, se realiza un análisis de las diferentes plataformas y aplicaciones basadas en la localización geográfica que actualmente existen. También se hace una comparativa de las diferentes tecnologías aplicables al proyecto. Por último se analiza la arquitectura de sistema que se ha seleccionado para el desarrollo de la aplicación.
3. Capítulo 3:
En el tercer capítulo se realiza una descripción detallada de los requisitos de usuario, de los requisitos funcionales, y de los casos de uso.
4. Capítulo 4
En el cuarto capítulo se describen las decisiones tomadas para la realización del diseño de la aplicación, y se detalla la implementación de la misma.
5. Capítulo 5
El capítulo 5 contiene las conclusiones del proyecto y las líneas futuras de desarrollo.
6. Anexo
En el capítulo dedicado para el anexo, se detalla cómo dar de alta una aplicación nueva en Facebook, así como los pasos necesarios para integrar la red social en una aplicación web desde cero.

GLOSARIO

Capitulo 2

Estado de la cuestión y estudio previo

2.1 Estado de la cuestión

En este capítulo se estudian y analizan los servicios basados en la localización y las diferentes redes sociales basadas en este servicio que actualmente existen, llevando a cabo una comparativa de las mismas. Además se analizan los diferentes gestores de bases de datos y servidores web que podemos encontrar en la actualidad con el fin de comprender los motivos de las decisiones técnicas tomadas a la hora de desarrollar el proyecto. Por último, se hace un repaso de la arquitectura del sistema que se ha utilizado.

2.2 Estudio Previo

2.2.1 Redes Sociales

¿Qué es una Red Social? Existen múltiples definiciones y teorías sobre los que son y no son las Redes Sociales, sin que haya aún consenso sobre las mismas. Se podría definir una red social como “un lugar en la red cuya finalidad es permitir a los usuarios relacionarse, comunicarse, compartir contenido y crear comunidades”

Según el Instituto Nacional de Tecnologías de la Comunicación (INTECO) en su “Estudio sobre la privacidad de los datos y la seguridad de la Información en las redes sociales online”, publicado en el 2009, definen las redes sociales como “los servicios prestados a través de Internet que permiten a los usuarios generar un perfil público, en el que plasmar datos personales e información de uno mismo, disponiendo de herramientas que permitan interactuar con el resto de usuarios afines o no al perfil publicado”. [INTECO]

Wikipedia define las redes sociales como “estructuras sociales compuestas por grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos”.

El inicio de las redes sociales se atribuye a Randy Conrads, con la creación del sitio web “classmates.com” [RANDY] para mantener el contacto entre ex compañeros del instituto. Desde entonces, se ha creado un gran boom, que ha desembocado en la creación de numerosas redes sociales así como el crecimiento de las funcionalidades puestas a disposición de los usuarios de las mismas.

Las Redes Sociales se suelen referir a las plataformas en Internet, cuyo propósito es facilitar la comunicación y otros temas sociales en los sitios web. Son páginas que permiten a las personas conectar con sus amigos, incluso realizar nuevas amistades, a fin de compartir contenidos, interactuar, crear comunidades sobre intereses similares: trabajo, lecturas, juegos, amistad, relaciones interpersonales. Las redes sociales en Internet han ganado su lugar de una manera vertiginosa, convirtiéndose en promisorios negocios para empresas, artistas, marcas, profesionales independientes y, sobre todo, en lugares para encuentros humanos.

Las redes sociales se pueden dividir según la tipología de las mismas de igual manera que los portales de Internet, en horizontales y en verticales:

Las redes sociales Horizontales serían aquellas que buscan proveer herramientas para la interrelación en general. Ejemplos de este tipo de redes serían Facebook, Tuenti, Google+

Las redes sociales Verticales se pueden dividir a su vez en función del tipo de usuario o del tipo de actividad a la que están dirigidas. Redes sociales como LinkedIn, dirigidas al sector profesional, se enmarcarían dentro de las redes sociales verticales por usuario. Un ejemplo de red social vertical por actividad sería la red social Twitter, que promueve una actividad concreta: el microblogging.

El uso de las redes sociales en España está bastante extendido entre los usuarios de internet. De hecho el 55% de los internautas españoles de 16 a 45 años declara acceder a diario a redes sociales o comunidades online. Este porcentaje aumenta hasta un 80% cuando atendemos a una frecuencia de acceso semanal. Además, únicamente un 7% declara no haber accedido en ninguna ocasión. [COCKTAIL]

Al igual que en el caso de la experiencia de acceso, la frecuencia de acceso a redes sociales a través del móvil se vincula también con el perfil de usuario de redes sociales, así aproximándose al 30% los usuarios que están accediendo varias veces en semana o diariamente.

Casi la totalidad de los usuarios de redes sociales disponen de un terminal avanzado, con acceso a Internet. Entre estos, hay una importante presencia de móviles de última generación, con posibilidad de descargar aplicaciones y acceder a Internet de forma más avanzada.

Una de las funcionalidades que nos llevan a este estudio es el uso de los Servicios Basados en la Localización en redes sociales como veremos en el siguiente apartado.

2.2.2 Servicios Basados en la Localización

Los Servicios Basados en la Localización (SBL o LBS, de su acrónimo inglés Location Based Services) son servicios que integran una localización o ubicación de un dispositivo con otra información para ofrecer un servicio personalizado a los usuarios basándose en la información de su ubicación geográfica.

El esquema típico de un Servicio Basado en la Localización, se compone de un Sistema de Información Geográfica (SIG o GIS de su acrónimo Geographic Information System), alguna tecnología de posicionamiento, como bien puede ser un módulo GPS o un Servicio de Posicionamiento suministrado por el operador de red, y por último de la tecnología de comunicación de redes que permita transferir información y hacerla accesible desde Internet.

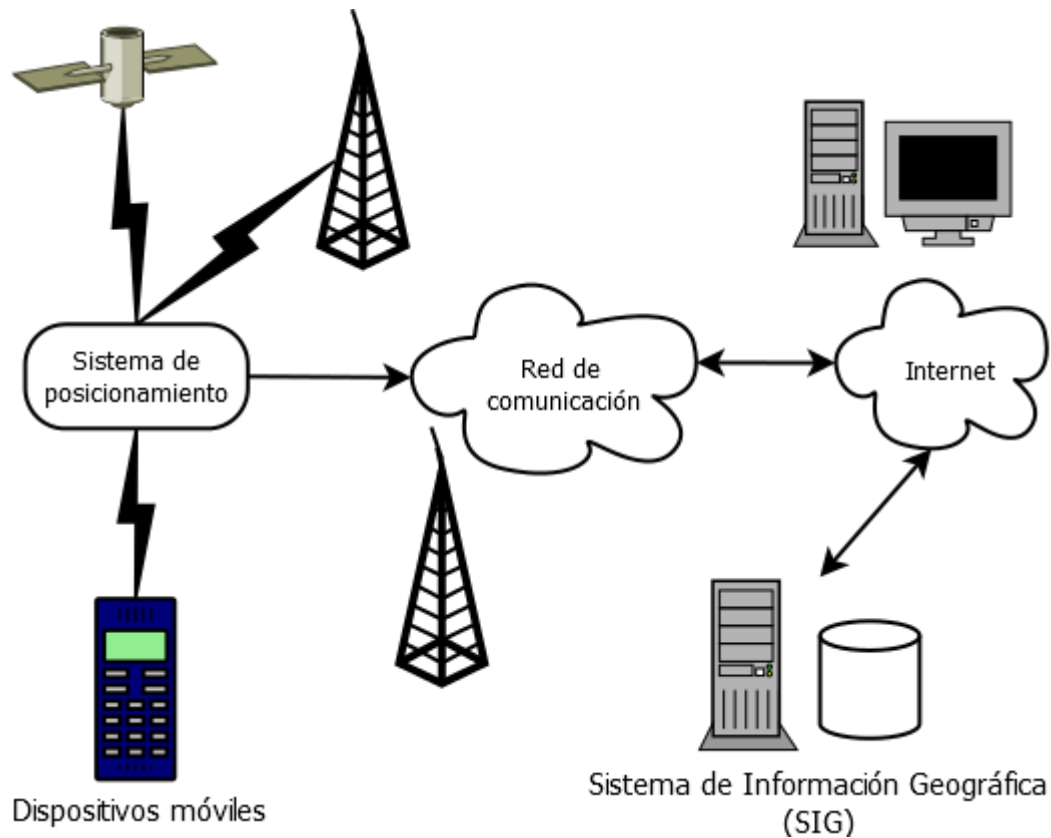


Ilustración 2 - Arquitectura típica de sistemas SBL

Se pueden diferenciar tres elementos principales cuya intersección compone un Sistema Basado en la Localización: Por un lado encontramos el Sistema de Información Geográfica, que se define como cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada; Las tecnologías de comunicación, que hacen que los dispositivos móviles puedan comunicarse con los diferentes componentes (satélites GPS, torres de comunicación...); y el tercer componente es Internet, que hace accesible toda la información desde cualquier lugar.

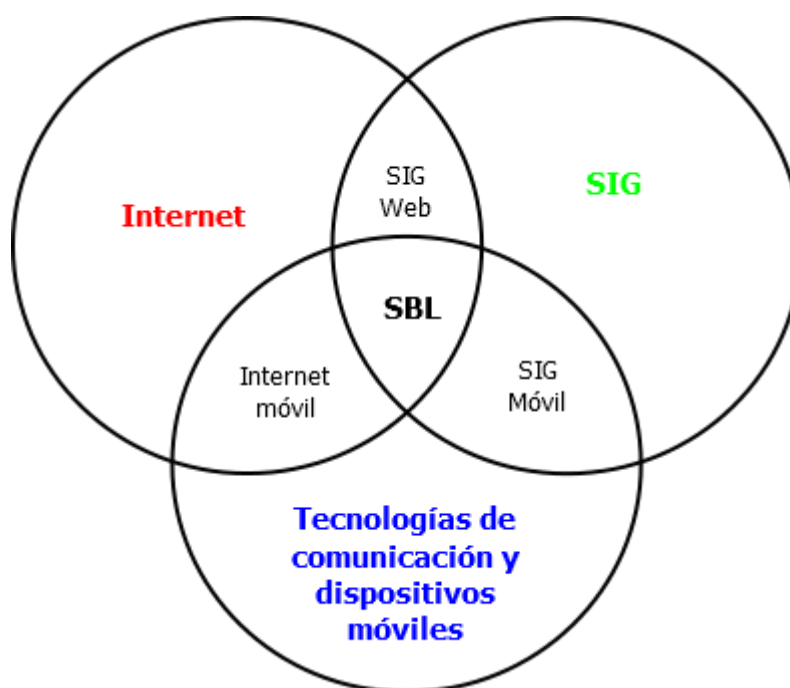


Ilustración 3 - Composición de sistemas Basados en la Localización

2.2.3 Redes sociales con Servicios Basados en la Localización

En los últimos años, el interés por los Servicios Basados en la Localización ha aumentado considerablemente. Hoy en día existen diferentes redes sociales con Servicios Basados en la Localización que proporcionan al usuario funcionalidades a partir de sus necesidades en tiempo real:

2.2.3.1 Foursquare

Foursquare es un Servicio Basado en Localización web aplicada a las redes sociales. Fue creada en 2009 por Dennis Crowley y Selvadurai Naveen. La idea principal de la red se basa en realizar “check-ins” en aquellos lugares en los que uno se encuentra, que pueden ser establecimientos, museos, lugares emblemáticos, etcétera, y compartir recomendaciones y opiniones acerca de los “check-ins” de tus contactos. Existen recompensas por visitar lugares y existe la posibilidad de adquirir puntos por “descubrir” nuevos lugares, de tal manera que al llegar a cierta cantidad de puntos, obtienes medallas (badges), que puedes compartir en distintas redes sociales como Facebook o Twitter. Todas las recompensas, puntos y medallas, son virtuales, y no sirven más que para competir con otros usuarios, quizás esta sea la clave por la cual esta red social ha alcanzado recientemente los 15 millones de usuarios y 1,5 mil millones de “check-ins”.

2.2.3.2 Gowalla

Gowalla es un Servicio Basado en Localización aplicada a las redes sociales que fue creada en 2007. El funcionamiento de esta red social es parecido al de Foursquare, con la diferencia de que no se centra tanto como un juego competitivo para alcanzar ciertos objetivos antes que tus amigos, sino que disimula este aspecto centrándose más en compartir información de localizaciones con tus amigos.

Una funcionalidad de Gowalla que es interesante es que permite crear recorridos que unen varios “check-ins” llamados “Trips”, adaptando el concepto de rutas temáticas que puedes recomendar a tus amigos.

El día 2 de diciembre de 2011 Gowalla es comprado por Facebook por aproximadamente 9 millones de euros.

[GOWALLA]

2.2.3.3 Google Latitude

Google Latitude es un Servicio Basado en Localización móvil desarrollado por Google que fue dado a conocer el 4 de febrero de 2009. Permite a un usuario que otras personas de su lista de contactos puedan rastrear dónde se encuentra, y viceversa. Se basa en la utilización de dispositivos móviles con acceso a Internet a través de los cuales se ubica al usuario, aunque también se puede hacer uso del servicio a través de navegadores convencionales. Otra ventaja del servicio es que las búsquedas que se realizan a través de su buscador se basan en la posición del usuario.

2.2.3.4 Facebook Places

Facebook Places es un servicio que se integra dentro de la red social Facebook. Nace el 18 de agosto de 2010 como una manera de competir directamente con Foursquare. Es un Servicio Basado en Localización móvil que permite realizar "check-ins" en aquellos lugares que deseas compartir con tus contactos de Facebook. En noviembre de 2010, Facebook incorporó una nueva funcionalidad llamada "Deals" (Ofertas) que ofrecen al usuario la posibilidad de obtener descuentos y promociones al realizar un "check-in" en bares, restaurantes, cafeterías y supermercados, de manera similar a si se utilizara una tarjeta de fidelización del lugar. En España, cabe destacar la promoción que se realizó a través de Facebook Places en la cadena de cafeterías Starbucks, mediante la cual, el usuario al realizar un "check-in" en alguna de las cafeterías seleccionadas, recibía de manera gratuita un café o un té, a cambio de promocionar el establecimiento en su Muro.

2.2.4 Sistemas gestores de Bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) o DBMA (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Otro objetivo es facilitar el manejo de transacciones, obteniendo independencia y consistencia de los datos. De esta manera, para el usuario es transparente cómo se almacenan físicamente los datos y garantiza que la información se encuentra segura para todos los usuarios y se respetan los permisos asignados.

Los SGBD permiten manipular grandes volúmenes de datos, y deben gestionar correctamente el acceso concurrente de varios usuarios a los mismos.

Se ha realizado un estudio de las principales características de tres de los SGBD más populares utilizados en aplicaciones Web. Los sistemas seleccionados para la comparación han sido Microsoft SQL Server, Oracle y MySQL, y a continuación se muestra una tabla comparativa con las principales características de las mismas y las ventajas y desventajas que han ayudado a elegir MySQL como SGBD para el proyecto.

2.2.4.1 SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

SQL Server es capaz de poner gran cantidad de datos simultáneamente a muchos usuarios, teniendo como principales ventajas la seguridad, la escalabilidad y la estabilidad del sistema. Admite una arquitectura Cliente/Servidor, que permite que las tareas se repartan entre los proveedores de servicios o recursos, llamados servidores, y los demandantes de dichos recursos, los clientes. Es un sistema que únicamente funciona bajo el sistema operativo de Microsoft, por lo que además del coste de la licencia por su uso, es necesario contar con un servidor con Windows.

2.2.4.2 Oracle

Oracle es un sistema de gestión de base de datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando el soporte de transacciones, la estabilidad, la escalabilidad y el soporte multiplataforma.

Para desarrollar en Oracle utilizamos PL/SQL un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos, también por norma general se suele utilizar SQL.

Hasta hace poco, su dominio en el mercado de servidores empresariales ha sido casi total, ya que es un sistema robusto y con características que garantizan la seguridad y la integridad de los datos, asegurando que las transacciones se ejecutan de manera correcta y sin causar inconsistencias.

A pesar de todo, el coste de las licencias para su uso es demasiado elevado.

2.2.4.3 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

Es un sistema ligero y relativamente rápido. Comparado con los otros sistemas puede ser algo escaso de características, aunque las principales como son escalabilidad, estabilidad y multiplataforma las cumple.

Cuando se utiliza un motor de Base de datos no transaccional, como es MyISAM, es una base de datos muy rápida en la lectura, y para entornos web, normalmente existe un baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en la lectura de datos, lo que hace que MySQL sea un sistema ideal para aplicaciones Web.

Por otra parte, MySQL se ofrece bajo una licencia GNU (GNU General Public License, GNU GPL), por lo que para cualquier uso sea compatible con esta licencia, el uso de la misma es gratuito. Para empresas que deseen incorporar MySQL en productos comerciales, deberán comprar una licencia específica para su uso.

2.2.4.4 Tabla comparativa

	SQL Server	MySQL	Oracle
Sistema Operativo	Microsoft Windows	Multiplataforma	Multiplataforma
Tipo de licencia	Miscrosoft EULA	Privada	GPL
Conexión cifrada	Sí	Sí	Sí
Escalabilidad	Sí	No	Sí
Estabilidad	Sí	No	Sí
Relacional	Sí	Sí	Sí
Transaccional	Sí	Sí	Sí
Multiusuario	Sí	Sí	Sí
Multihilo	Sí	Sí	Sí
Disparadores	Sí	Sí	Sí
Claves primarias	Sí	Sí	Sí
Claves ajenas	Sí	Sí	Sí
Vistas	Sí	Sí	Sí
Procedimientos Almacenados	Sí	Sí	Sí

Ilustración 4 - Tabla Comparativa SGBD

2.2.4.5 Conclusiones

MySQL cumple de perfectamente con los requisitos necesarios para el correcto funcionamiento de la aplicación desarrollada. Además, el ser software de libre distribución y sin coste alguno para el proyecto, ha sido la característica principal de la elección. Por último, MySQL cuenta con una comunidad inmensa de desarrolladores, que pueden aportar una solución rápida en caso de producirse algún error en el sistema gestor de base de datos.

2.2.5 Servidores Web

Un servidor web o servidor HTTP es un programa que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones.

El término Servidor Web también se aplica para referirse a la máquina física en la cual está funcionando este software, y cuyo propósito es proveer los datos que los ordenadores cliente le solicitan.

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).

Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

Existen en el mercado distintas opciones a la hora de elegir un programa Servidor Web. Los más importantes son Apache, Internet Information Server (IIS) y Cherokee. Otros servidores, destacan por su simpleza y rapidez, como Lighthttpd y Thttpd, aunque no van a ser objeto de este estudio.

A continuación se presenta una breve descripción de cada Programa Servidor Web con sus características principales, así como una tabla resumen de las mismas, que explican de manera resumida las razones por las cuales se ha elegido Apache como Servidor Web para la aplicación.

2.2.5.1 Apache

Es un servidor Web HTTP de código abierto, para plataformas UNIX, Windows y Macintosh. Apache presenta entre otras, características altamente configurables, bases de datos de autenticación y negociado de contenido, aunque un punto en su contra es la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el

GLOSARIO

70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. [NETCRAFT]

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web de forma segura y confiable. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

2.2.5.2 Internet Information Server (IIS)

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, como por ejemplo, Microsoft Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

2.2.5.3 Cherokee

Cherokee es un servidor web multiplataforma. Su objetivo es ser rápido y completamente funcional, sin dejar de ser liviano comparado con otros servidores web. Está escrito completamente en C. Puede usarse como un sistema embebido y soporta complementos para aumentar sus funcionalidades. Es software libre, disponible bajo la Licencia Pública General de GNU. Actualmente es desarrollado y mantenido por una comunidad abierta de desarrolladores.

2.2.5.4 Tabla comparativa

	Apache	IIS	Cherokee
Sistema Operativo	Multiplataforma	Microsoft Windows	Multiplataforma
Software Libre	Sí	No	Sí
Tipo de licencia	Licencia Apache compatible con GPL versión3	Microsoft EULA	GPL GNU
Diseño Modular / Soporte de Plug-ins	Sí	Sí	Sí
Conexiones permanentes	Sí	Sí	Sí
Escala a Servidores SMP	Sí	Sí	Sí
Soporte Virtual Servers	Sí	Sí	Sí
CGI's	Sí	Sí	Sí
Páginas de error personalizadas	Sí	Sí	Sí
Conexiones seguras	Sí	Sí	Sí

HTTPS			
-------	--	--	--

Ilustración 5 - Tabla Comparativa Servidores Web

2.2.5.5 Conclusiones

Hoy en día, el servidor Web Apache es el servidor más usado de Internet, con una utilización del 65% aproximadamente. Las principales metas de su diseño son: velocidad, simplicidad, multiplataforma y facilidad del desarrollo distribuido. En todas estas características se encuentra por encima de sus competidores. Éstas y los más de 15 años de andadura del proyecto, han sido las principales razones para elegir Apache como Servidor Web de la aplicación.

GLOSARIO

Capítulo 3

Análisis

3.1 Análisis del sistema

Una vez se han descrito el estado de la cuestión y el entorno tecnológico se definirán las funcionalidades requeridas para el sistema. Se realizará una descripción general del sistema como idea aproximada, concretando luego con los requisitos de usuario, generando los casos de uso y por último definiendo los requisitos software de sistema.

3.1.1 Descripción general del sistema

El objetivo de este sistema es implementar una red social de geolocalización basada en Facebook y Google Maps. Para ello, será necesario disponer de ambas APIs y su documentación con la que nos podamos comunicar y realizar las peticiones necesarias.

La red social se basa en compartir lugares en un mapa, usando Facebook como plataforma social, por ello la aplicación deberá incluir todas las interfaces necesarias para que el usuario pueda hacer uso de todas las funcionalidades de la red social. Estas funcionalidades son:

- Registro de nuevos usuarios usando la cuenta de Facebook
- Creación de nuevo perfil
- Modificación de perfil
- Registro de nueva posición sin estado

GLOSARIO

- Registro de nueva posición con estado
- Registro de nueva posición con estado combinado con actualización de estado en Facebook
- Modificación de la posición actual
- Vista del histórico de posiciones
- Creación de comentario en la posición actual
- Creación de comentario en la posición actual combinado con actualización de estado en Facebook
- Vista de la posición actual de amigos
- Vista del histórico de posiciones de amigos
- Creación de comentarios en posiciones de amigos
- Creación de comentarios en posiciones de amigos combinado con actualización de estado en Facebook
- Búsqueda de direcciones en el mapa
- Acceso directo a amigos en el mapa

Además se deberá mostrar cualquier error que se produzca durante el uso de la aplicación

3.1.2 Requisitos de usuario

Los requisitos de usuario son los requisitos que el cliente final describe, en este caso el alumno, y se describen en este apartado con el siguiente formato de tabla:

Identificador	Nombre del requisito		
Fuente	Origen del que proviene el requisito	Prioridad	Importancia del requisito. Puede tomar los valores Alta, Media, Baja
Descripción	Texto descriptivo del requisito		

Tabla 2 - Formato de tabla de especificación de requisitos

3.1.2.1 Requisitos de restricción

RU-01	Aplicación de escritorio		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación se diseñará para ser ejecutada principalmente en un entorno de escritorio y deberá ser compatible con la mayoría de los navegadores actuales.		

Tabla 3 - RU-01

RU-02	Utilización de mapas de Google		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación deberá utilizar la tecnología que suministra Google para mostrar los mapas y direcciones: Google Maps		

Tabla 4 - RU-02

RU-03	Utilización de localización geográfica del navegador		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación deberá ser capaz de solicitar al navegador información sobre la posición geográfica del usuario y de procesar esa información mostrándola en un mapa.		

Tabla 5 - RU-03

RU-04	Aplicación Facebook		
Fuente	Cliente	Prioridad	Alta

GLOSARIO

Descripción	La aplicación se desarrollará usando el SDK que proporciona Facebook para que forme parte de Facebook Apps.
-------------	---

Tabla 6 - RU-04

RU-05	Credenciales de usuario Facebook		
Fuente	Cliente	Prioridad	Alta
Descripción	Para poder usar e instalar la aplicación, será necesario que el usuario tenga una cuenta de Facebook activa y operativa.		

Tabla 7 - RU-05

RU-06	Credenciales de usuario		
Fuente	Cliente	Prioridad	Alta
Descripción	Únicamente usuarios que hayan accedido a su cuenta de Facebook, y hayan instalado la aplicación, podrán acceder a la misma		

Tabla 8 - RU-06

RU-07	Almacenamiento de credenciales		
Fuente	Cliente	Prioridad	Alta
Descripción	El sistema almacenará las credenciales del usuario y sólo será necesario instalar la aplicación la primera vez.		

Tabla 9 - RU-07

RU-08	Interfaz en castellano		
Fuente	Cliente	Prioridad	Media
Descripción	La aplicación tendrá la interfaz en castellano		

Tabla 10 - RU-08

RU-09	Protección de datos personales		
Fuente	Cliente	Prioridad	Media
Descripción	La aplicación tendrá mecanismos de protección para que sólo el usuario pueda acceder a su información personal.		

Tabla 11 - RU-09

RU-10	Amigos del usuario		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación deberá mostrar únicamente aquellos amigos de Facebook que hayan instalado la aplicación.		

Tabla 12 - RU-10

3.1.2.2 Requisitos de Interfaz

RU-11	Buscar direcciones y mostrar los resultados en el mapa		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación proporcionará un cuadro de búsqueda donde el usuario podrá introducir direcciones. La aplicación deberá ser capaz de traducir esas direcciones en puntos geográficos en el mapa.		

Tabla 13 - RU-11

RU-12	Mostrar la posición actual del usuario		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación deberá ser capaz bajo demanda del usuario de proporcionar información aproximada de su posición en forma de texto, además de gráficamente en el mapa.		

Tabla 14 - RU-12

RU-13	Mostrar la posición de los amigos del usuario		
Fuente	Cliente	Prioridad	Alta
Descripción	La aplicación deberá mostrar en el mapa gráficamente la posición de los amigos en Facebook que hayan instalado la aplicación y proporcionado su posición. Además proporcionará información extra en forma de texto de la posición y opcionalmente de su estado.		

Tabla 15 - RU-13

RU-14	Actualizar los datos del mapa		
Fuente	Cliente	Prioridad	Media

GLOSARIO

Descripción	La aplicación proporcionará un mecanismo para actualizar los datos mostrados en el mapa. Esto será útil para ver si un amigo ha cambiado su posición en el mapa.
-------------	--

Tabla 16 - RU-14

RU-15	Configurar la aplicación		
Fuente	Cliente	Prioridad	Media
Descripción	La aplicación dará la posibilidad de configurar de manera predeterminada la publicación de nuevas posiciones en Facebook, o la utilización de la localización geográfica por parte del navegador.		

Tabla 17 - RU-15

RU-16	Recomendar la aplicación		
Fuente	Cliente	Prioridad	Baja
Descripción	La aplicación dará la posibilidad de recomendar la aplicación a otros amigos que no la tengan ya instalada		

Tabla 18 - RU-16

RU-17	Cerrar la sesión		
Fuente	Cliente	Prioridad	Baja
Descripción	La aplicación dará la posibilidad de salir de la sesión actual y cerrar la aplicación		

Tabla 19 – RU-17

RU-18	Registrar nueva posición		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá registrar una nueva posición en el mapa, bien usando la posición geográfica que proporciona el navegador Web, o bien seleccionando una posición en el mapa de manera manual. Además podrá incluir un texto con información sobre qué hace en esa posición (a partir de ahora “estado”)		

Tabla 20 – RU-18

RU-19	Posibilidad de publicar nueva posición en Facebook		
Fuente	Cliente	Prioridad	Alta

GLOSARIO

Descripción	Un usuario al guardar una nueva posición en el mapa, tendrá la posibilidad de publicar la posición en Facebook a través de una actualización de su estado.
-------------	--

Tabla 21 - RU-19

RU-20	Actualizar posición		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá actualizar el estado de su última posición		

Tabla 22 - RU-20

RU-21	Ver histórico de posiciones		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá ver un histórico de sus últimas posiciones en el mapa y navegar por ellas.		

Tabla 23 - RU-21

RU-22	Ver histórico de posiciones de amigos		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá ver un histórico de las últimas posiciones de sus amigos en el mapa y navegar por ellas.		

Tabla 24 - RU-22

RU-23	Comentar posiciones		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá realizar comentarios sobre su posición en el mapa o sobre la de sus amigos.		

Tabla 25 - RU-23

RU-24	Publicar comentarios sobre posiciones en Facebook		
Fuente	Cliente	Prioridad	Alta
Descripción	Un usuario podrá publicar actualizaciones en su perfil sobre los comentarios que realice de su posición o sobre las posiciones de sus amigos.		

Tabla 26 - RU-24

3.1.3 Casos de Uso

En este apartado se describen los casos de uso de manera gráfica y textual. Los casos de uso describen un uso del sistema y cómo el sistema interactúa con el usuario, describiendo de manera clara la funcionalidad del sistema.

3.1.3.1 Diagrama

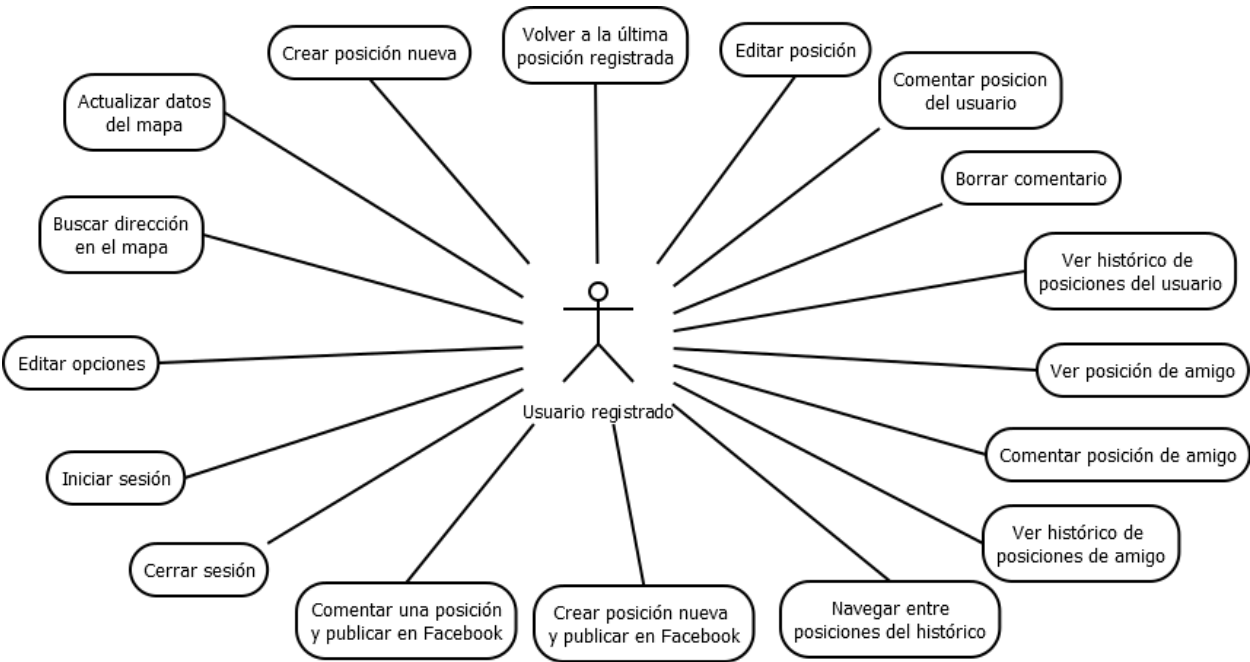


Ilustración 6 - Diagrama de Casos de Uso (Usuario registrado)

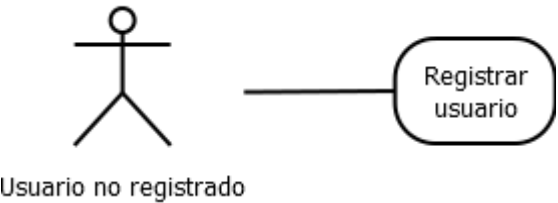


Ilustración 7 - Diagrama de Casos de Uso (Usuario no registrado)

3.1.3.2 Descripción textual

A continuación se muestra el formato de tabla que se va a utilizar para describir de manera textual cada caso de uso:

Identificador	Nombre del caso de uso
Actores	Lista de actores involucrados en el caso de uso
Objetivo	Meta a conseguir con el caso de uso
Precondiciones	Condiciones que se deben cumplir para que el caso de uso se pueda llevar a cabo
Postcondiciones	Hechos que se han de cumplir si el flujo de eventos normal se ha ejecutado correctamente
Escenario básico	Secuencia numerada de acciones principales de la interacción entre los actores y el sistema para lograr el objetivo.

Tabla 27 - Formato de tabla de especificación de casos de uso

CU-01	Registrar usuario
Actores	Usuario no registrado
Objetivo	Asociar una cuenta de usuario en Facebook con la aplicación
Precondiciones	El usuario debe tener una cuenta de Facebook activa y operativa y haber iniciado sesión en la misma El usuario no ha vinculado previamente su cuenta de Facebook con la aplicación
Postcondiciones	El usuario vincula su cuenta de Facebook con la aplicación y puede acceder a la misma
Escenario básico	El usuario accede a la aplicación El usuario hace clic en el botón <i>Conectar</i> El usuario acepta conectar su cuenta de Facebook con la aplicación El usuario acepta opcionalmente los permisos adicionales para usar la aplicación que solicita Facebook El sistema da de alta la nueva cuenta y le asigna el identificador de usuario que proporciona Facebook

Tabla 28 – CU-01

GLOSARIO

CU-02	Iniciar sesión
Actores	Usuario registrado
Objetivo	Acceder a la aplicación
Precondiciones	El usuario ha accedido previamente a su cuenta de Facebook El usuario está dado de alta en el sistema y su cuenta de Facebook asociada.
Postcondiciones	El usuario ha iniciado sesión
Escenario básico	El usuario accede a la aplicación El usuario hace clic sobre el botón <i>Conectar</i> Se inicia sesión

Tabla 29 – CU-02

CU-03	Cerrar sesión
Actores	Usuario registrado
Objetivo	Cerrar la sesión de usuario en el sistema
Precondiciones	El usuario ha iniciado sesión
Postcondiciones	Se cierra la sesión de usuario
Escenario básico	El usuario hace clic sobre el botón <i>Desconectar</i> del panel superior de la aplicación El sistema elimina la sesión de usuario Se cierra la sesión de usuario

Tabla 30 – CU-03

CU-04	Editar opciones
Actores	Usuario registrado
Objetivo	Modificar las opciones de la aplicación disponibles
Precondiciones	El usuario ha iniciado sesión
Postcondiciones	El usuario modifica las opciones de la aplicación

GLOSARIO

Escenario básico	<p>El usuario hace clic sobre el botón <i>Configuración</i> del panel superior de la aplicación</p> <p>El usuario modifica la opción deseada</p> <p>El usuario hace clic sobre el botón <i>Guardar</i></p> <p>El sistema actualiza las opciones en su base de datos</p> <p>Las opciones de aplicación se modifican</p>
------------------	--

Tabla 31 – CU-04

CU-05	Buscar dirección en el mapa
Actores	Usuario registrado
Objetivo	Encontrar una dirección y mostrarla en el mapa
Precondiciones	El usuario ha iniciado sesión
Postcondiciones	Si la dirección existe se muestra en el mapa, si no existe se muestra un mensaje de error
Escenario básico	<p>El usuario hace clic en el cuadro de búsqueda del panel superior de la aplicación</p> <p>El usuario introduce la dirección a buscar</p> <p>El usuario hace clic sobre el botón <i>buscar</i> o pulsa <i>Intro</i></p> <p>El sistema envía la petición al servidor de Google</p> <p>Se muestra en el mapa el resultado de la búsqueda</p>

Tabla 32 – CU-05

CU-06	Actualizar datos del mapa
Actores	Usuario registrado
Objetivo	Actualizar los datos mostrados en el mapa, tanto la posición actual del usuario como de sus amigos
Precondiciones	El usuario ha iniciado sesión
Postcondiciones	Se muestra la posición actual del usuario y de sus amigos en el mapa

GLOSARIO

Escenario básico	<p>El usuario hace clic en el botón <i>Refrescar Mapa</i> del panel superior de la aplicación</p> <p>El sistema hace una petición al servidor para obtener la última posición almacenada de los amigos del usuario</p> <p>El sistema hace una petición al navegador para obtener la posición geográfica del usuario, o si no está disponible, al servidor con la última posición almacenada</p> <p>Se muestra la posición actual del usuario y sus amigos en el mapa</p> <p>Se muestra en el panel derecho los comentarios sobre la posición actual del usuario si las hubiera.</p>
------------------	---

Tabla 33 – CU-06

CU-07	Crear posición nueva
Actores	Usuario registrado
Objetivo	Almacenar una nueva posición en el mapa
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>La posición a almacenar es diferente a la última posición almacenada</p> <p>El usuario ha realizado una búsqueda en el mapa, o el usuario ha movido su posición manualmente, o el navegador ha posicionado al usuario en una posición diferente.</p>
Postcondiciones	El sistema almacena una nueva posición del usuario
Escenario básico	<p>El sistema muestra en el mapa una chincheta de color rojo y sobre ella un globo con la información geográfica de la posición</p> <p>El usuario informa opcionalmente el cuadro llamado <i>¿qué haces aquí?</i> del globo mostrado</p> <p>El usuario desmarca la opción <i>Publicar en Facebook</i> del globo mostrado</p> <p>El usuario hace clic sobre el enlace <i>Guardar</i> del globo mostrado</p> <p>La aplicación envía los datos al servidor y son almacenados</p> <p>El sistema actualiza el mapa con la nueva información</p>

Tabla 34 – CU-07

CU-08	Volver a la última posición registrada
Actores	Usuario registrado
Objetivo	Mostrar en el mapa la última posición registrada en el sistema
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha realizado una búsqueda en el mapa, o el usuario ha movido su posición manualmente, o el navegador ha posicionado al usuario en una posición diferente.</p>
Postcondiciones	Se muestra en el mapa la última posición almacenada por el sistema
Escenario básico	<p>El usuario hace clic sobre el botón <i>Tu posición</i> del panel superior de la aplicación</p> <p>En el panel lateral, el usuario hace clic sobre el enlace <i>Volver a tu última posición</i></p> <p>El sistema hace una petición al servidor para obtener la última posición almacenada del usuario</p> <p>Se muestra en el mapa la última posición almacenada del usuario</p> <p>Se muestra en el panel lateral los comentarios sobre la posición si los hubiera.</p>

Tabla 35 – CU-08

CU-09	Editar posición
Actores	Usuario registrado
Objetivo	Editar el texto de la última posición almacenada en el sistema
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El sistema está mostrando la última posición almacenada</p>
Postcondiciones	El texto de la última posición se modifica

GLOSARIO

Escenario básico	<p>El sistema muestra en el mapa una chincheta de color rojo y sobre ella un globo con la información geográfica de la posición</p> <p>El usuario hace clic sobre el enlace <i>Editar</i> en el globo</p> <p>Se activa el cuadro de texto con el texto que se hubiera salvado con la posición</p> <p>El usuario edita el texto</p> <p>El usuario hace clic en el enlace <i>Guardar</i> del globo</p> <p>El sistema actualiza el texto de la posición</p>
------------------	--

Tabla 36 – CU-09

CU-10	Comentar posición de usuario
Actores	Usuario registrado
Objetivo	Realizar un comentario en una posición propia almacenada en el sistema
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha seleccionado su última posición</p>
Postcondiciones	El usuario realiza un comentario en una de sus posiciones almacenadas en el sistema
Escenario básico	<p>El usuario hace clic en el enlace <i>Comentar</i> en el panel lateral</p> <p>En el panel lateral, se activa un cuadro de texto</p> <p>El usuario escribe un texto en el cuadro de texto en el panel lateral</p> <p>El usuario hace clic en el enlace <i>Guardar</i></p> <p>El sistema almacena el comentario</p> <p>El panel lateral se actualiza con el nuevo comentario</p>

Tabla 37 – CU-10

CU-11	Borrar comentario
Actores	Usuario registrado
Objetivo	Eliminar un comentario que el propio usuario ha realizado

GLOSARIO

Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha realizado las acciones necesarias para mostrar una posición almacenada en el sistema en la que ha realizado un comentario</p>
Postcondiciones	El usuario elimina su comentario del sistema
Escenario básico	<p>El usuario hace clic sobre el enlace <i>Eliminar</i> del panel lateral del comentario que desea eliminar</p> <p>El sistema muestra el texto <i>Eliminando</i> y envía la petición de borrado del comentario</p> <p>Se actualiza el panel lateral y el comentario eliminado ya no se muestra</p>

Tabla 38 – CU-11

CU-12	Ver histórico de posiciones del usuario
Actores	Usuario registrado
Objetivo	Mostrar en el mapa las últimas posiciones almacenadas del usuario
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha salvado previamente al menos 2 posiciones</p> <p>El usuario ha seleccionado ver su última posición</p>
Postcondiciones	El sistema muestra en el mapa las últimas posiciones en forma de chinchetas identificadas con una letra del alfabeto.
Escenario básico	<p>El usuario hace clic en el enlace <i>Más posiciones</i></p> <p>La aplicación actualiza el mapa mostrando las últimas posiciones del usuario</p> <p>El panel lateral se actualiza para mostrar un panel de navegación entre las posiciones del histórico</p>

Tabla 39 – CU-12

CU-13	Ver posición de amigo
Actores	Usuario registrado
Objetivo	Mostrar en el mapa las posiciones actuales de los amigos de Facebook que usan la aplicación

GLOSARIO

Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario cuenta con al menos un amigo de Facebook que ha instalado la aplicación y ha almacenado su posición en el sistema</p>
Postcondiciones	<p>El sistema muestra en el mapa la posición del amigo seleccionado en forma de chincheta de color rojo, y un globo con la información de la misma.</p>
Escenario básico	<p>El usuario hace clic sobre el botón <i>Tus amigos</i></p> <p>El panel lateral se actualiza para mostrar un panel de navegación con las fotografías del perfil de los amigos del usuario</p> <p>El usuario hace clic sobre la imagen de uno de sus amigos</p> <p>El mapa se actualiza para mostrar la última posición del usuario y abrir un globo con la información de la misma.</p> <p>El panel lateral se actualiza para mostrar los comentarios de la posición del amigo seleccionado.</p>

Tabla 40 – CU-13

CU-14	Comentar posición de amigo
Actores	Usuario registrado
Objetivo	Realizar un comentario sobre la posición seleccionada de un amigo
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha seleccionado la posición de uno de sus amigos</p>
Postcondiciones	El usuario realiza un comentario sobre la posición de uno de sus amigos

GLOSARIO

Escenario básico	<p>El usuario selecciona el enlace <i>Comentar</i> en el panel lateral</p> <p>En el panel lateral, se activa un cuadro de texto</p> <p>El usuario escribe un texto en el cuadro de texto en el panel lateral</p> <p>El usuario hace clic en el enlace <i>Guardar</i></p> <p>El sistema almacena el comentario</p> <p>El panel lateral se actualiza con el nuevo comentario</p>
------------------	--

Tabla 41 – CU-14

CU-15	Ver histórico de posiciones de amigo
Actores	Usuario registrado
Objetivo	Mostrar en el mapa las últimas posiciones registradas de un amigo del usuario
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha seleccionado la posición de uno de sus amigos</p> <p>El amigo seleccionado tiene al menos 2 posiciones almacenadas en el sistema</p>
Postcondiciones	El sistema muestra en el mapa las últimas posiciones almacenadas en el sistema
Escenario básico	<p>El usuario hace clic en el enlace <i>Más posiciones</i> del globo</p> <p>La aplicación actualiza el mapa mostrando las últimas posiciones del amigo seleccionado</p> <p>El panel lateral se actualiza para mostrar un panel de navegación entre las posiciones del histórico</p>

Tabla 42 – CU-15

CU-16	Navegar entre posiciones del histórico
Actores	Usuario registrado
Objetivo	Navegar entre las distintas posiciones almacenadas de un usuario en el sistema

GLOSARIO



Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha seleccionado ver el histórico de posiciones de un usuario (o de sí mismo)</p> <p>Se muestra en el mapa la última posición almacenada en el sistema del histórico de posiciones, llamada A</p> <p>El histórico de posiciones tiene almacenadas dos posiciones, A y B (que es anterior a la posición A)</p>
Postcondiciones	El usuario navega entre las distintas posiciones almacenadas a lo largo del tiempo de un usuario
Escenario básico	<p>El usuario hace clic sobre el botón  del globo</p> <p>El sistema muestra la posición B en el mapa, y los comentarios de la posición B en el panel lateral</p> <p>El usuario hace clic sobre el botón  del globo</p> <p>El sistema muestra la posición A en el mapa, y los comentarios de la posición A en el panel lateral.</p>

Tabla 43 – CU-16

CU-17	Crear una posición nueva y publicar en Facebook
Actores	Usuario registrado
Objetivo	Almacenar una nueva posición en el mapa y publicar una actualización de estado en Facebook
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>La posición a almacenar es diferente a la última posición almacenada</p> <p>El usuario ha realizado una búsqueda en el mapa, o el usuario ha movido su posición manualmente, o el navegador ha posicionado al usuario en una posición diferente.</p>
Postcondiciones	<p>El sistema almacena una nueva posición del usuario</p> <p>El sistema realiza una actualización de estado en Facebook con la información de la posición almacenada</p>

GLOSARIO

Escenario básico	<p>El sistema muestra en el mapa una chincheta de color rojo y sobre ella un globo con la información geográfica de la posición</p> <p>El usuario informa opcionalmente el cuadro llamado <i>¿qué haces aquí?</i> del globo mostrado</p> <p>El usuario marca la opción <i>Publicar en Facebook</i> del globo mostrado</p> <p>El usuario hace clic sobre el enlace <i>Guardar</i> del globo mostrado</p> <p>La aplicación envía los datos al servidor y son almacenados</p> <p>El sistema se conecta con Facebook y publica en el nombre del usuario una actualización del sistema con la información de la posición</p> <p>El sistema actualiza el mapa con la nueva información</p>
------------------	--

Tabla 44 – CU-17

CU-18	Comentar una posición y publicar en Facebook
Actores	Usuario registrado
Objetivo	Realizar un comentario en una posición almacenada en el sistema y realizar una actualización de estado en Facebook sobre el comentario de la posición del usuario
Precondiciones	<p>El usuario ha iniciado sesión</p> <p>El usuario ha seleccionado una posición almacenada en el sistema (su última posición, una posición de su histórico, la última posición de un amigo, o una del histórico de un amigo)</p> <p>El sistema muestra en el mapa la posición seleccionada</p>
Postcondiciones	El usuario realiza un comentario en una posición almacenada en el sistema y publica una actualización en Facebook sobre ese comentario

GLOSARIO

Escenario básico	<p>El usuario hace clic en el enlace <i>Comentar</i> en el panel lateral</p> <p>En el panel lateral, se activa un cuadro de texto</p> <p>El usuario escribe un texto en el cuadro de texto en el panel lateral</p> <p>El usuario marca la opción <i>Publicar</i></p> <p>El usuario hace clic en el enlace <i>Guardar</i></p> <p>El sistema almacena el comentario</p> <p>El sistema accede a Facebook para publicar una actualización en el perfil del usuario cuya posición ha comentado</p> <p>El panel lateral se actualiza con el nuevo comentario</p>
------------------	--

Tabla 45 – CU-18

3.1.4 Requisitos software del sistema

Partiendo de los requisitos definidos por el cliente, y los casos de uso, se definen los requisitos software del sistema que formarán la base para conocer qué va a hacer la aplicación.

3.1.4.1 Requisitos funcionales

RF-01	Alta de un nuevo usuario		
Fuente	RU-04, RU-05	Prioridad	Alta
Descripción	El sistema podrá dar de alta nuevos usuarios, utilizando como base los datos proporcionados por Facebook, especialmente su identificador de usuario único.		

Tabla 46 - RF-01

RF-02	Inicio de sesión		
Fuente	RU-04, RU-06, RU-07	Prioridad	Alta
Descripción	Los usuario registrados en el sistema, podrán abrir sesión en la aplicación haciendo uso de su cuenta de Facebook		

Tabla 47 - RF-02

RF-03	Cierre de sesión		
Fuente	RU-17	Prioridad	Alta
Descripción	Los usuario que hayan accedido a la aplicación, podrán en cualquier momento cerrar su sesión activa y salir del sistema.		

Tabla 48 - RF-03

RF-04	Registrar una nueva posición en el mapa		
Fuente	RU-01, RU-02, RU-03, RU-11, RU-18	Prioridad	Alta

GLOSARIO

Descripción	El usuario podrá añadir una nueva posición y el sistema deberá almacenar esa posición en la base de datos. La posición geográfica se obtendrá a través del servicio de geoposicionamiento que proporcione el navegador web utilizado por el usuario, mediante una búsqueda de un lugar o aquella posición geográfica que el usuario indique al interactuar con el mapa. Los datos a almacenar serán latitud, longitud, y opcionalmente un texto describiendo la posición que proporcionará el usuario.
-------------	--

Tabla 49 - RF-04

RF-05	Mostrar posición actual		
Fuente	RU-12, RU-14	Prioridad	Alta
Descripción	El sistema deberá mostrar la posición geográfica del usuario en el mapa		

Tabla 50 - RF-05

RF-06	Mostrar amigos		
Fuente	RU-13	Prioridad	Alta
Descripción	El sistema deberá mostrar la posición geográfica de los amigos de Facebook que hayan instalado la aplicación		

Tabla 51 - RF-06

RF-07	Mostrar histórico de posiciones		
Fuente	RU-21, RU-22	Prioridad	Alta
Descripción	El sistema deberá mostrar a petición del usuario el histórico de posiciones de un amigo o de sí mismo. Debido a esto, el sistema deberá ser capaz de almacenar en la base de datos un registro con las posiciones de cada usuario		

Tabla 52 - RF-07

GLOSARIO

RF-08	Realizar comentarios en las posiciones		
Fuente	RU-23	Prioridad	Alta
Descripción	El usuario podrá realizar comentarios sobre posiciones geográficas, tanto de sus amigos como suyas. El sistema deberá almacenar esos comentarios, y enlazarlos a la entrada de la posición geográfica correspondiente. El sistema también almacenará comentarios de posiciones del histórico.		

Tabla 53 - RF-08

RF-09	Mostrar comentarios en las posiciones		
Fuente	RU-23	Prioridad	Alta
Descripción	El usuario podrá consultar los comentarios que se han realizado sobre sus posiciones o sobre las de sus amigos.		

Tabla 54 - RF-09

RF-10	Realizar publicaciones en Facebook		
Fuente	RU-19, RU-24	Prioridad	Alta
Descripción	El sistema usará la API que proporciona Facebook para publicar el contenido que opcionalmente el usuario desee compartir en la red social. El usuario podrá optar por compartir posiciones o comentarios sobre posiciones.		

Tabla 55 - RF-10

RF-11	Buscar un lugar en el mapa		
Fuente	RU-11	Prioridad	Alta
Descripción	El usuario podrá realizar búsquedas de direcciones y el sistema mostrará el resultado en el mapa. Para realizar estas búsquedas se usarán los mecanismos que proporciona la API de Google.		

Tabla 56 - RF-11

GLOSARIO

RF-12	Configuración de la aplicación		
Fuente	RU-15	Prioridad	Media
Descripción	La aplicación tendrá algunos parámetros que podrán ser configurables por el usuario.		

Tabla 57 - RF-12

3.1.4.2 Requisitos no funcionales

RNF-01	Uso de la API de Google Maps		
Fuente	RU-02	Prioridad	Alta
Descripción	El sistema hará uso de la API que proporciona Google para mostrar mapas, posiciones en el mapa, cálculos de direcciones a latitud-longitud y a la inversa: API de codificación geográfica (versión 3)		

Tabla 58 - RNF-01

RNF-02	Uso de la API de Facebook		
Fuente	RU-04, RU-05	Prioridad	Alta
Descripción	El sistema hará uso de la API que proporciona Facebook, así de los scripts necesarios para que la aplicación desarrollada pueda ser utilizada por el cliente con sus credenciales de Facebook.		

Tabla 59 - RNF-02

RNF-03	Interfaz en castellano		
Fuente	RU-08	Prioridad	Alta
Descripción	La aplicación tendrá la interfaz en castellano		

Tabla 60 - RNF-03

RNF-04	Aplicación de escritorio		
Fuente	RU-01	Prioridad	Alta
Descripción	La aplicación se diseñará y desarrollará para ser utilizada como aplicación de escritorio a través de un navegador web compatible.		

Tabla 61 - RNF-04

RNF-05	Cuenta en Facebook		
Fuente	RU-05	Prioridad	Alta
Descripción	Para poder instalar la aplicación, será imprescindible que el cliente posea una cuenta en Facebook activa y operativa.		

Tabla 62 - RNF-05

RNF-06	Datos protegidos		
Fuente	RU-09	Prioridad	Alta
Descripción	El sistema tendrá mecanismos para que cada usuario pueda consultar únicamente sus datos de posición, así como los del sus amigos en Facebook, en ningún caso podrá tener acceso a datos de otras personas.		

Tabla 63 - RNF-06

Capítulo 4

Diseño e implementación

4.1 Diseño e implementación

Una vez se han definido los requisitos tanto funcionales como no funcionales, en este capítulo se analizan las decisiones tomadas para la realización del diseño de la aplicación, que servirán al desarrollador de guía para una correcta implementación.

4.1.1 Metodología de desarrollo

La metodología usada para estructurar, planear y controlar el proceso de desarrollo del proyecto se ha basado en el modelo incremental o de desarrollo iterativo y creciente.

Este modelo nace como respuesta a las debilidades del modelo tradicional en cascada, en el que el inicio de cada etapa debe esperar a la finalización de la etapa anterior, y cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. En este modelo, en cambio, se planifica el proyecto en diversos bloques temporales, llamados iteraciones, donde se repite un determinado proceso de trabajo que incluye pruebas para poder cumplir con los objetivos marcados. En cada iteración, el producto se va evolucionando dependiendo de las iteraciones anteriores completadas, en función de los requisitos marcados por el cliente.

La idea principal de este modelo es desarrollar un sistema de manera incremental, permitiendo al desarrollador obtener ventaja de lo aprendido a lo largo del desarrollo anterior, incrementando versiones entregables del sistema. Los pasos clave en el proceso son comenzar con una implementación simple de los requisitos del sistema, e iterativamente mejorar cada versión hasta que el sistema completo esté implementado. En cada iteración se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

Este tipo de modelo es muy útil en el desarrollo de aplicaciones que impliquen diseños de interfaces de usuario, ya que es muy complicado que no surjan problemas de usabilidad desde el principio, y en cada iteración se pueden ir corrigiendo estos problemas hasta llegar al diseño final. Otra ventaja de este modelo es que permite separar la complejidad del proyecto, gracias a su desarrollo por partes de cada iteración.

A continuación se definen las tareas del ciclo de vida incremental:

Etapa de inicialización, donde se crea una versión inicial del sistema con el fin de que el cliente pueda tener una idea principal e inicial del producto, que le permita interactuar con la herramienta, y se pueda conocer a su vez de primera mano las necesidades y problemas encontrados.

Etapa de iteración, donde a partir de la información obtenida en fases anteriores, se rediseña e implementa de nuevo. En esta etapa se vuelve a obtener retroalimentación del cliente y se analizan las funcionalidades disponibles del sistema, y se realizan las iteraciones necesarias hasta alcanzar las metas fijadas en la lista de control de proyecto.

4.1.2 Arquitectura del sistema

Las aplicaciones web permiten a los usuarios acceder a la información proporcionada por la aplicación en tiempo real, y realizar modificaciones de los datos compartidos en el servidor Web, normalmente almacenando primero los datos en el navegador Web, y enviándolos a petición del cliente al servidor Web para ser procesados.

Este comportamiento se conoce como arquitectura Cliente-Servidor, donde el cliente mantiene una copia de los datos reales en local, donde realiza las modificaciones sobre los datos pertinentes, y se envían dichos datos modificados al servidor. De esta manera los datos no se intercambian con los demás clientes, evitando conflictos.

La arquitectura Cliente-Servidor es un modelo de aplicación distribuida en la que la capacidad de proceso está repartida entre los proveedores de servicios o recursos, llamados Servidores, y los demandantes de esos servicios o recursos, los Clientes.

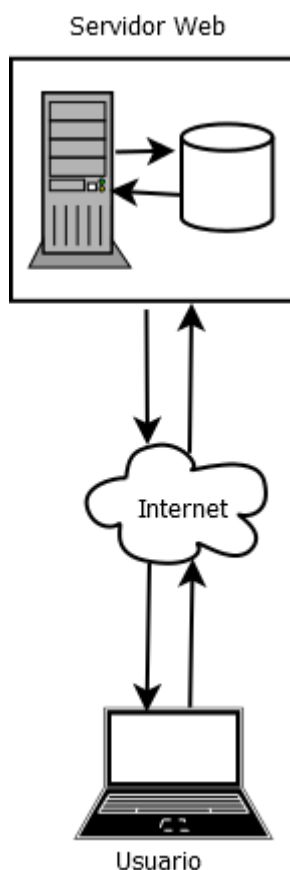


Ilustración 8 - Diagrama de arquitectura Cliente-Servidor

La figura anterior muestra de manera simplificada la arquitectura Cliente-Servidor para un servicio Web, donde un cliente, inicia las peticiones o solicitudes, manteniendo un papel activo en la comunicación. El cliente esperará y recibirá las respuestas del servidor, el cual a su vez realiza un papel pasivo en la comunicación, esperando recibir solicitudes de los clientes. Una vez recibe alguna solicitud del cliente, se procesa y envía de vuelta al cliente. El Servidor por lo general está diseñado para poder procesar las solicitudes de muchos clientes (modelo típico de servidor Web, donde muchos clientes se conectan a una página Web alojada en un Servidor)

GLOSARIO

El sistema que se va a desarrollar, se basará en esta arquitectura, introduciendo otros elementos necesarios para el correcto funcionamiento de la aplicación.

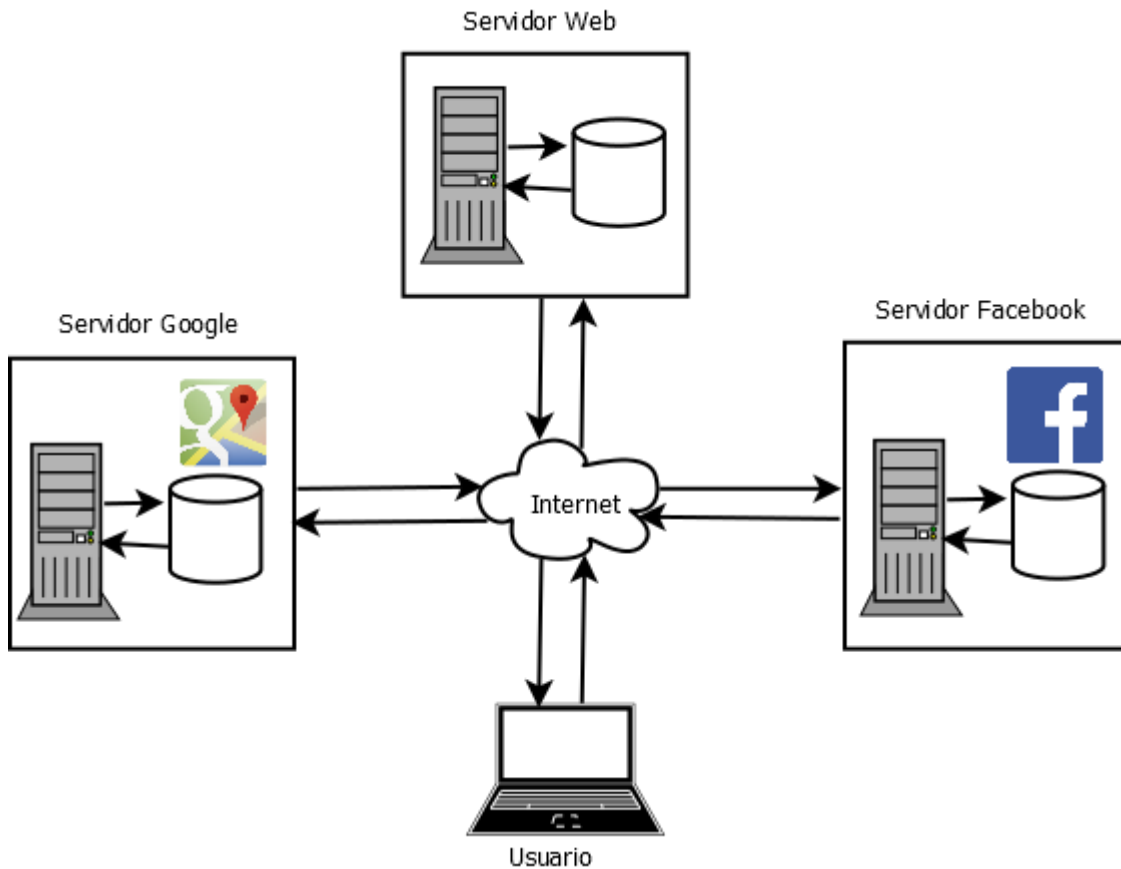


Ilustración 9 - Diagrama de la arquitectura seleccionada

La aplicación a desarrollar estará alojada en el servidor Web. El cliente accederá a la aplicación a través de un navegador Web estándar, realizando peticiones al Servidor Web. A su vez el servidor Web de Google, será el encargado de proporcionar información geográfica al servidor, y de información gráfica al cliente, es decir, tanto el usuario como el servidor Web actuarán como clientes del servidor Web de Google. Por último, el servidor de Facebook servirá para que el usuario pueda conectarse a su cuenta de Facebook, y proporcione al servidor Web acceso a los datos del usuario dentro de la red social.

Tanto el servidor Google, como el servidor Facebook descritos en la figura anterior actúan como una caja negra, ya que para el sistema es independiente cómo estén implementados internamente, o el número de servidores que compone cada elemento.

A continuación se detalla brevemente el funcionamiento y los elementos que componen la Plataforma Facebook a través de la cual la aplicación será capaz de leer y escribir datos.

4.1.2.1 Plataforma Facebook

La plataforma Facebook proporciona una interfaz entre nuestra aplicación y Facebook y es esencial para poder crear una aplicación dentro de la red social.

GLOSARIO

El modelo típico de una aplicación Web es aquel en el cual nuestra aplicación y sus datos están alojados en un Servidor Web, y los clientes mediante de sus navegadores, acceden a la misma a través de Internet.

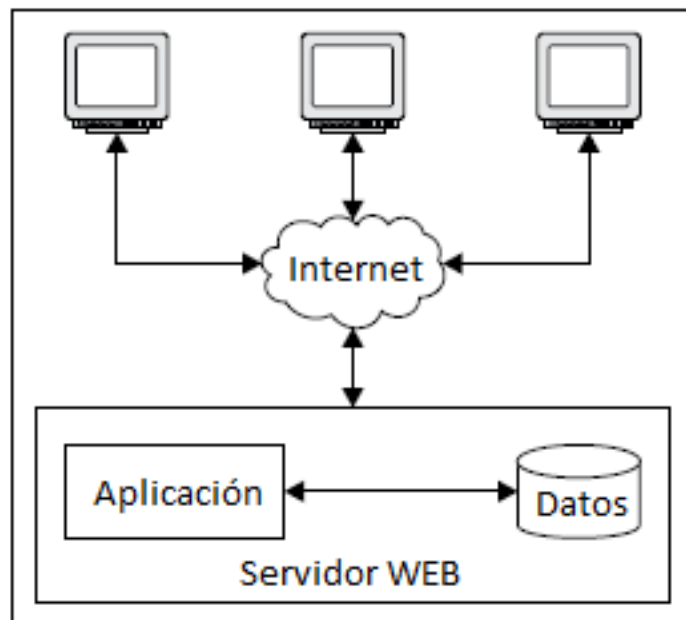


Ilustración 10 - Modelo típico de Aplicación Web

El modelo de Aplicación Web de Facebook es un poco diferente, ya que permite a los usuarios acceder a través de Facebook a nuestra aplicación, pero ésta nunca estará alojada en un servidor de Facebook, sino en nuestro servidor.

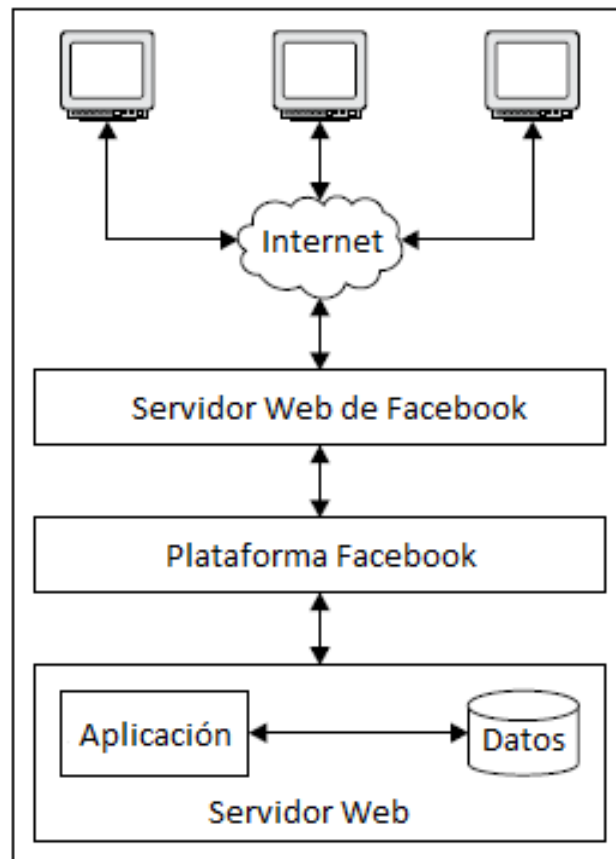


Ilustración 11 - Modelo de Aplicación Web de Facebook

La plataforma de Facebook está compuesta por dos elementos principales:

- API de Facebook (API, del acrónimo en inglés Application Programming Interface, o Interfaz de Programación de Aplicaciones). Es un conjunto de librerías de software que permite trabajar con una aplicación (en este caso Facebook), sin tener que conocer nada sobre cómo funciona internamente.
- FQL, o Facebook Query Language (Lenguaje de Consulta de Facebook). Los datos de Facebook se almacenan en Bases de Datos, por lo que se utiliza este lenguaje propio de Facebook para acceder a los mismos. Permite consultar tablas de la base de datos de Facebook directamente sin tener que recurrir a un método de la API.

4.1.3 Modelo E/R

El modelo E/R o Entidad-Relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.

En el modelo E/R se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades:

- **Entidad.**- Objeto del mundo real sobre el que queremos almacenar información. Las entidades están compuestas de *atributos* que son los datos que definen el objeto. De entre los atributos habrá uno o un conjunto de ellos que no se repite; a este atributo o conjunto de atributos se le llama *clave* de la entidad. En toda entidad siempre hay al menos una clave que en el peor de los casos estará formada por todos los atributos de la tabla. Ya que pueden haber varias claves y necesitamos elegir una, lo haremos atendiendo a estas normas:
 - **Que sea única.**
 - **Que se tenga pleno conocimiento de ella.**
 - **Que sea mínima**, ya que será muy utilizada por el gestor de base de datos.
- **Relación.**- Asociación entre entidades, sin existencia propia en el mundo real que estamos modelando, pero necesaria para reflejar las interacciones existentes entre entidades. Las relaciones pueden ser de tres tipos:
 - **Relaciones 1-1.**- Las entidades que intervienen en la relación se asocian una a una.
 - **Relaciones 1-n.**- Una ocurrencia de una entidad está asociada con muchas (n) de otra.
 - **Relaciones n-n.**-Cada ocurrencia, en cualquiera de las dos entidades de la relación, puede estar asociada con muchas (n) de la otra y viceversa.

Las entidades se representan gráficamente mediante rectángulos, y su nombre, identificativo y unívoco, se muestra en su interior. Las propiedades de las entidades se representan mediante bolas que cuelgan de las entidades o de las relaciones a las que pertenece. Las relaciones se representan gráficamente mediante rombos con su nombre en el interior.

[BASESDATOS]

A continuación se muestra el diagrama Entidad-Relación de la aplicación:

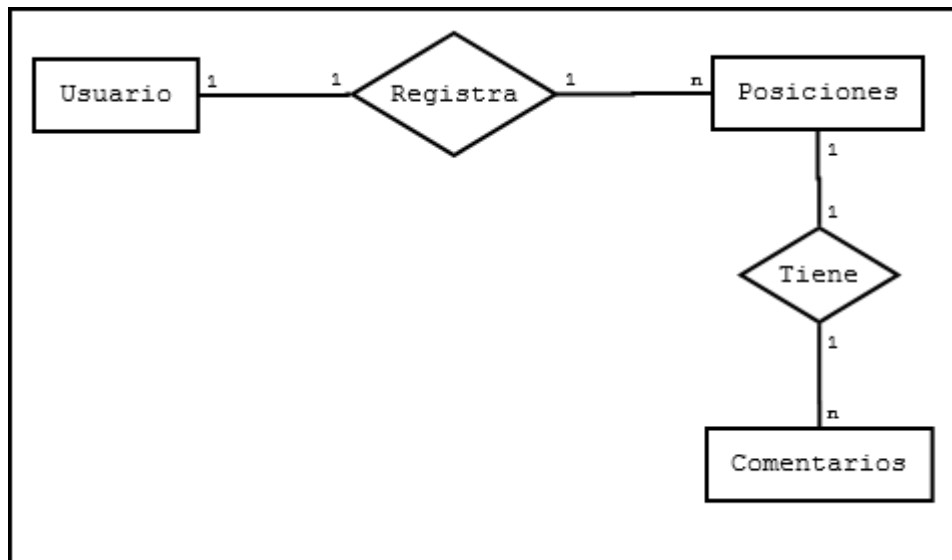


Ilustración 12 - Diagrama Entidad-Relación

El sistema almacena información relevante del usuario, que es único. La aplicación permite a los usuarios registrar posiciones del mapa en la base de datos, tantas como desee. Cada posición se considera como única, y pertenece a un único usuario. Para una posición concreta, la aplicación permite realizar comentarios sobre esa posición almacenada, los usuarios pueden realizar tantos comentarios como deseen.

4.1.4 Modelo Relacional

El modelo relacional constituye una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que, además de proporcionar los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de un álgebra relacional) para su manipulación, sin ambigüedad posible.

El modelo relacional se basa en el concepto matemático de relación. En este modelo, la información se representa en forma de “tablas” o relaciones, donde cada fila de la tabla se interpreta como una relación ordenada de valores (un conjunto de valores relacionados entre sí).

Una relación tiene un nombre, que debe ser único dentro de la relación, un conjunto de atributos que representan sus propiedades, y un conjunto de tuplas que incluyen los valores que cada uno de los atributos toma para cada elemento de la relación. Además, los atributos podrán tomar un conjunto de valores válidos, también llamado dominio.

[A.SILBERSCHATZ] [D.CUADRA]

GLOSARIO

A continuación se muestra el esquema del modelo relacional de la aplicación:

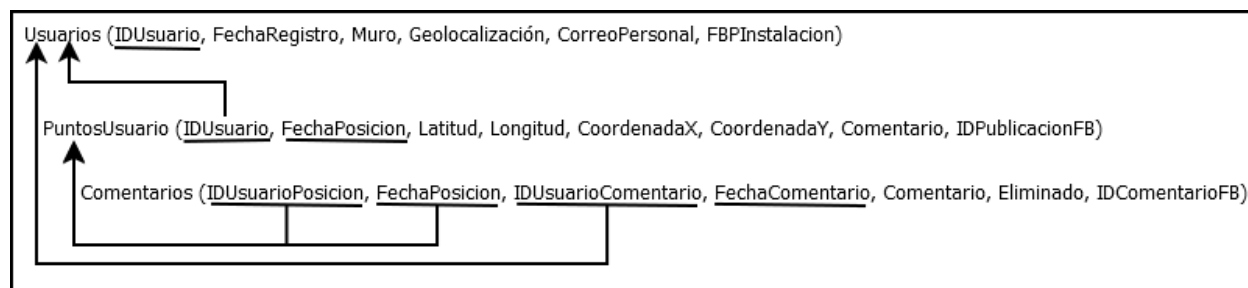


Ilustración 13 – Esquema del Modelo Relacional de la aplicación

Las relaciones se muestran con un identificador único, seguido entre paréntesis por los atributos que contendrá cada tupla de esa relación. Los atributos de cada relación están separados por comas.

Aquellos atributos que son clave primaria de la tabla aparecen subrayados, y los que son clave ajena de otra tabla aparecen con una flecha apuntando a la tabla a cuyo atributo está siendo referenciado.

A continuación se describe detalladamente cada relación:

Tabla **Usuarios**: Esta tabla contiene la información relacionada con los usuarios de la aplicación. Será consultada por el sistema para dar de alta nuevos usuarios, consultar los ya existentes, o consultar y actualizar sus preferencias de uso de la aplicación. Cada usuario tiene un identificador único, es por eso que el campo IDUsuario es la clave primaria.

- **IDUsuario**: Clave primaria de la tabla. Indica el identificador único de usuario de la aplicación, que coincide con el identificador numérico del usuario en Facebook. Es tipo entero, y puede tener una longitud de hasta 20 caracteres numéricos. Debido a que es clave primaria, no admite valores nulos.
- **FechaRegistro**: Este campo contiene la fecha de registro del usuario en el sistema en formato unix. Por lo tanto es un tipo entero, que admite hasta 11 caracteres. No admite valores nulos
- **Muro**: Indica a la aplicación si por defecto viene marcada o no la opción de publicar actualizaciones de estado en Facebook al salvar una nueva posición. Es un tipo bit, que no admite nulos y por defecto tiene valor 1.
- **Geolocalización**: Indica a la aplicación si por defecto utiliza o no los mecanismos que proporciona el navegador web para posicionar geográficamente al usuario. Es tipo bit, no admite nulos y por defecto toma el valor 1.
- **CorreoPersonal**: Este campo almacena el correo personal que el usuario indica en las opciones de la aplicación. Es tipo varchar, de hasta 255 caracteres, y puede ser nulo.
- **FBPIstalacion**: Este campo almacena el identificador único de la publicación en Facebook en el caso de que el usuario haya indicado al instalar la aplicación su deseo de publicar y promocionar este hecho en Facebook. En caso contrario almacena el valor 0. Es un tipo varchar, de hasta 255 caracteres, no admite nulos.

Tabla **PuntosUsuario**: Esta tabla contiene toda la información relacionada con los puntos geográficos que los usuarios han registrado en el sistema a través de la aplicación. El sistema consultará esta tabla para obtener la última posición almacenada del usuario, en forma de latitud-longitud, y en ocasiones en forma de coordenadas X-Y. Además el sistema también consultará esta tabla para obtener el histórico de posiciones de un usuario concreto, o las posiciones actuales de un amigo. La clave primaria de esta tabla viene dada por la unión de los campos IDUsuario y IDPosicion, ya que un usuario puede registrar “n” posiciones, pero todas con diferente fecha. Los campos que definen esta tabla son los siguientes:

- **IDUsuario**: Este campo es clave ajena de la tabla Usuarios. Forma parte de la clave primaria, junto con el campo FechaPosicion. Es de tipo entero, admite hasta 20 caracteres y no puede ser nulo.
- **FechaPosicion**: Este campo junto con IDUsuario, forma parte de la clave primaria de la tabla. Es de tipo entero y almacena la fecha de registro de la posición en formato unix. Tiene un tamaño máximo de 11 caracteres numéricos y no puede ser nulo.
- **Latitud**: Este campo indica el valor en formato decimal de la latitud del punto que registra el usuario en la aplicación. Puede tomar valores positivos y negativos. Es de tipo varchar, de hasta 15 caracteres. No admite nulos.
- **Longitud**: Este campo indica el valor en formato decimal de la longitud del punto que registra el usuario en la aplicación. Puede tomar valores positivos y negativos. Es de tipo varchar, de hasta 15 caracteres. No admite nulos.
- **CoordenadaX**: Este campo indica el valor en formato de coordenada X-Y, la coordenada X del punto que registra el usuario en la aplicación. Es de tipo entero, de hasta 11 caracteres, y no puede ser nulo.
- **CoordenadaY**: Este campo indica el valor en formato de coordenada X-Y, la coordenada Y del punto que registra el usuario en la aplicación. Es de tipo entero, de hasta 11 caracteres, y no puede ser nulo.
- **Comentario**: Este campo almacena el valor del texto que opcionalmente el usuario publica sobre la posición que registra. Además este texto coincide con el texto que se añade a la publicación en Facebook de la actualización de estado si el usuario lo desea. Es de tipo varchar, de hasta 255 caracteres y puede tomar valores nulos.
- **IDPublicacionFB**: Este campo almacena el identificador único de la publicación de la posición en Facebook como actualización de estado del usuario. Es de tipo varchar, de hasta 255 caracteres. En el caso de que no se realice publicación en Facebook, el campo toma valor 0. No admite nulos.

Tabla **Comentarios**: Esta tabla almacena la información de los comentarios que se realizan sobre las posiciones de cada usuario. El sistema accederá a esta tabla cada vez que se consulte la información sobre la posición de un usuario, para cargar los comentarios asociados a la misma. Los campos que definen esta tabla son los siguientes:

- **IDUsuarioPosicion**: Este campo es clave ajena de la tabla Posiciones. Forma parte de la clave primaria, junto con el campo FechaPosicion, IDUsuarioComentario y FechaComentario. Es de tipo entero, admite hasta 20

caracteres y no puede ser nulo. Contiene el identificador del usuario de cuya posición se comenta.

- **FechaPosicion:** Este campo junto con IDUsuarioPosicion, IDUsuarioComentario y FechaComentario, forma parte de la clave primaria de la tabla. Es de tipo entero y almacena la fecha de registro de la posición a la que pertenece el comentario en formato unix. Tiene un tamaño máximo de 11 caracteres numéricos y no puede ser nulo.
- **IDUsuarioComentario:** Este campo es clave ajena de la tabla Usuarios. Forma parte de la clave primaria, junto con el campo IDUsuarioPosicion, FechaPosicion y FechaComentario. Es de tipo entero, admite hasta 20 caracteres y no puede ser nulo. Contiene el identificador del usuario que realiza el comentario.
- **FechaComentario:** Este campo junto con IDUsuarioPosicion, FechaPosicion y IDUsuarioComentario, forma parte de la clave primaria de la tabla. Es de tipo entero y almacena la fecha de registro del comentario de la posición en formato unix. Tiene un tamaño máximo de 11 caracteres numéricos y no puede ser nulo.
- **Comentario:** Este campo almacena el comentario que se realiza de la posición. Es de tipo varchar, de hasta 255 caracteres, y no puede ser nulo.
- **Eliminado:** Este campo indica si el comentario ha sido eliminado y por lo tanto la aplicación no lo mostrará. Es de tipo bit, no admite nulos, y por defecto toma el valor 0.
- **IDComentarioFB:** Este campo almacena el identificador único del comentario de la publicación de la posición en Facebook como actualización de estado del usuario. Es de tipo varchar, de hasta 255 caracteres. En el caso de que no se realice publicación en Facebook, el campo toma valor 0. No admite nulos

4.1.5 Implementación

La implementación de la aplicación se ha llevado a cabo a partir del diseño propuesto. Se han mezclado varios lenguajes: HTML, JavaScript y CSS para la presentación de los datos, PHP, y SQL para el procesamiento y almacenamiento de los datos, con la ayuda de la técnica de desarrollo web AJAX; y por otra parte se ha hecho uso de librerías externas para conectar la herramienta con Facebook y Google Maps.

A continuación se hace una descripción de los detalles necesarios para comprender los aspectos más relevantes del desarrollo.

4.1.5.1 Librerías externas

Antes de entrar de lleno a describir cómo se ha implementado la aplicación, es necesario conocer cómo la herramienta hace uso de las librerías externas necesarias para integrar Facebook y Google Maps en el sistema.

4.1.5.1.1 SDK de Facebook para PHP

La Plataforma Facebook incluye un Kit de Desarrollo de Software o SDK (de las siglas en inglés Software Development Kit) para poder desarrollar aplicaciones para Facebook. Este kit, proporciona un conjunto de funcionalidades del lado del servidor para realizar llamadas a la API de Facebook, y así poder leer y escribir datos en la red social. El SDK que se ha utilizado para el desarrollo de la aplicación, es específico para el lenguaje PHP, y es necesario combinar su funcionalidad con el SDK de Facebook para JavaScript, que proporciona las funcionalidades del lado del cliente y completan al SDK para PHP.

El Kit de desarrollo contiene algunos scripts de ejemplo útiles para comprender junto con la documentación que proporciona Facebook en su sitio Web de desarrolladores el funcionamiento de la API.

Las funcionalidades del SDK están implementadas en dos scripts, que son los que deben ser incluidos siempre que se necesite tener acceso a los datos alojados en Facebook. Estos dos scripts están contenidos en la carpeta src y se llaman base_Facebook.php y Facebook.PHP.

La clase Facebook está definida en el script Facebook.PHP. Esta clase proporciona una implementación concreta que utiliza las sesiones de PHP para almacenar identificadores de usuario y tokens de acceso. Esta clase será la que utilizará la aplicación para poder comunicarse e interactuar con Facebook utilizando las llamadas a la API (que se definen en la clase BaseFacebook que se explica a continuación).

En el script base_Facebook.PHP se definen las clases BaseFacebook y FacebookApiException.

La clase BaseFacebook proporciona a nuestra aplicación acceso a la Plataforma Facebook. Contiene la mayoría de las funcionalidades necesarias para ello, así como acceso a la API pública de Facebook. Un listado completo de los métodos disponibles se puede encontrar en la siguiente URL

[API_FACEBOOK]

Siempre que sea necesario hacer uso del SDK, será necesario incluir el fichero Facebook.PHP en el script PHP que vaya a hacer uso del mismo. En nuestro caso, el SDK está alojado en la carpeta Facebook-Facebook-PHP-sdk/src/Facebook.PHP, por lo que será necesario añadir el siguiente código:


```
include_once('facebook-facebook-php-sdk/src/facebook.php');
```

Ilustración 14 - Incluir SDK de Facebook para PHP

Para instanciar un nuevo objeto de la clase Facebook, es necesario disponer de un identificador de la aplicación, y una clave secreta. Estos datos los proporciona Facebook al dar de alta una nueva aplicación en la página de desarrolladores. En la aplicación desarrollada, estos datos, junto con las credenciales para la conexión a la base de datos del servidor, se definen en un fichero de configuración y se deben cargar de la misma manera que el SDK de Facebook. Por lo tanto, para instanciar un nuevo objeto de la clase Facebook, se utilizará el siguiente script:

```
include_once 'lib/config.php';
include_once('facebook-facebook-php-sdk/src/facebook.php');

$facebook = new Facebook(array(
    'appId' => $GLOBALS['app_id'],
    'secret' => $GLOBALS['api_secret']
));
```

Ilustración 15 - Instanciar objeto Facebook

Una vez se ha instanciado el objeto Facebook, ya es posible obtener datos del usuario en Facebook. Para ello, primero es necesario que el usuario se haya autenticado en Facebook, y haya permitido a nuestra aplicación el acceso a sus datos. Este proceso de autenticación y permisos, se realiza combinando el SDK de Facebook para PHP y para JavaScript.

El cliente realiza una petición al servidor, que comprueba si se ha realizado con éxito. Con el siguiente script, se puede comprobar si el usuario ha hecho login en Facebook y ha dado permisos a nuestra aplicación para que acceda a sus datos:

```

include_once 'lib/config.php';
include_once('facebook-facebook-php-sdk/src/facebook.php');

global $facebook, $user, $user_profile, $logoutUrl, $loginUrl;

$facebook = new Facebook(array(
    'appId' => $GLOBALS['app_id'],
    'secret' => $GLOBALS['api_secret']
));

//En $user tenemos el ID del usuario que ha hecho login
$user = $facebook->getUser();

//Comprobamos si el usuario que ha hecho login tiene un token de acceso para la aplicación
if ($user) {
    try {
        //Para comprobar si el login es correcto en la aplicación, intentamos realizar una operacion
        //que solo esta permitida si se ha dado acceso a la aplicación
        $user_profile = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        //Se ha hecho login en Facebook, pero no en la aplicación
        error_log($e);
        $user = null;
    }
}

if ($user) {
    $logoutUrl = $facebook->getLogoutUrl();
} else {
    $loginUrl = $facebook->getLoginUrl();
}

```

Ilustración 16 - Script de comprobación de login en Facebook

En el caso de que el cliente no haya accedido a Facebook, o no haya permitido previamente a la aplicación a utilizar sus datos de Facebook, se cargará en el navegador web del cliente la librería JavaScript del SDK de Facebook, con las funcionalidades necesarias para realizar el login en Facebook y posibilitar al usuario aceptar los permisos de uso por parte de la herramienta de sus datos en Facebook.

El siguiente código JavaScript carga la librería del SDK de Facebook para JavaScript, inicializa el objeto FB, y comprueba si el usuario se ha autenticado y ha dado permisos a la aplicación, y proporciona las funciones necesarias para ello en caso negativo:

```
<script type="text/javascript" src="http://connect.facebook.net/en_US/all.js"></script>
```

Ilustración 17 - Código HTML para cargar la librería JavaScript del SDK de Facebook

```

<script type='text/javascript'>
    var fbuserid, fbtoken, appid = 1234567890, loggedin = false;

    FB.init({appId: appid, status: true, cookie: true, xfbml: true});
    FB.Event.subscribe('auth.sessionChange', function(response) {
        if (response.session) {
            var session = FB.getSession();
            fbtoken = session.access_token;
            fbuserid = session.uid;
        }
    });
    FB.Event.subscribe('auth.login', function(response) {
        window.location.reload();
    });
    FB.getLoginStatus(function(response) {
        if (response.session) {
            var session = FB.getSession();
            fbtoken = session.access_token;
            fbuserid = session.uid;
            window.location = loginUrl;
        }
    });
    function loginFB() {
        FB.login(function(response) {
            if (response.session) {
                var session = FB.getSession();
                fbtoken = session.access_token;
                fbuserid = session.uid;
            }
        }, {perms: 'publish_stream'});
    }
</script>

```

Ilustración 18 - Código JavaScript que complementa el login en Facebook

En el caso de que el usuario de la aplicación haya realizado correctamente el login en Facebook, y haya dado los permisos pertinentes a la aplicación para acceder a su información, la herramienta tendrá los accesos necesarios para realizarlo. Facebook utiliza para ello tokens de acceso. Los token de acceso son cadenas de texto opacas que identifican a un usuario, aplicación o página de Facebook y son utilizadas para realizar llamadas a la API de Facebook (graph API). El token incluye información acerca de cuándo el token expira y qué aplicación ha generado ese token. La mayoría de las llamadas a métodos de la API de Facebook necesitan realizarse acompañadas de un token de acceso válido, con el fin de conservar la privacidad de los datos de cada usuario. La siguiente figura muestra el flujo para obtener un token de acceso por parte de la aplicación.

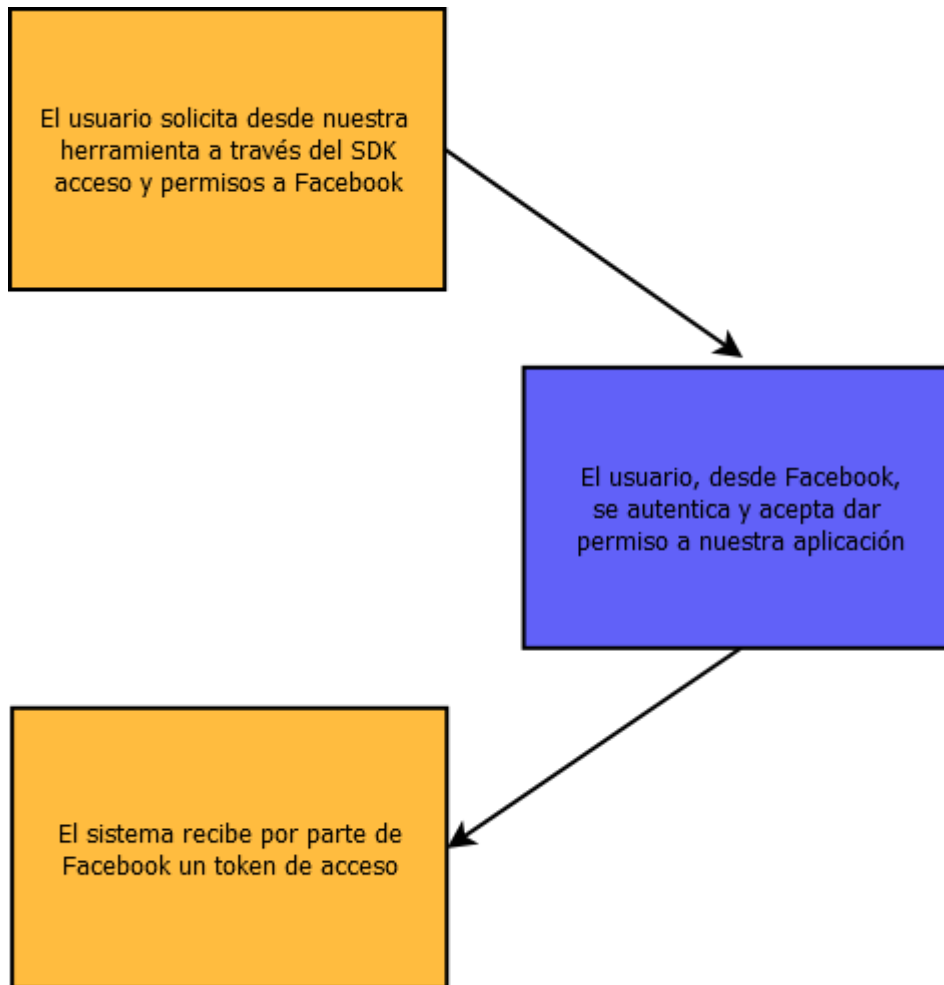


Ilustración 19 - Diagrama de solicitud de token de acceso

Nuestra aplicación, a través de la interacción del usuario, realiza una petición de acceso a Facebook. El usuario, desde Facebook, se autentica (si no lo estuviera ya) y da permisos a la aplicación. Facebook genera un token de acceso, y lo envía a nuestra aplicación, donde es almacenado para su uso con la API.

Una vez la aplicación ha almacenado el token de acceso, puede realizar llamadas a la API de Facebook. A continuación se muestra el código de ejemplo de uso de la API para publicar un nuevo comentario en el tablón del usuario en Facebook:

```
$message = $user_profile['name'].' ya comparte su posicion en "Donde Estas?";

$args = array(
    'message' => $message,
    'name' => utf8_encode('Dónde Estás?'),
    'link' => $callback_url,
    'source' => $callback_url,
    'picture' => $callback_url.'images/logotipo_cabecera.png'
);

$res = $facebook->api("/me/feed", "post", $args);
```

Ilustración 20 - Ejemplo de llamada a la API de Facebook para PHP

4.1.5.1.2 API JavaScript de Google Maps

Para la inserción de mapas en nuestra aplicación, se ha utilizado el API de JavaScript de Google Maps. El API proporciona diversas utilidades para manipular mapas y para añadir contenido en el mapa mediante diversos servicios.

Esta API que proporciona Google es gratuita, y para hacer uso de ella, es necesario cargar la API mediante una clave de API. El uso de esta clave proporciona a Google estadísticas de uso de la misma, que en caso de superar los límites de uso gratuito (25000 peticiones por día), se deberá obtener una clave de API con un coste asociado.

Para poder obtener una clave de API, es necesario contar con una cuenta de Google operativa y seguir las instrucciones detalladas en la página de documentación de la API versión 3 de Google Maps

[API_GOOGLE]

Una vez se ha obtenido la clave de API, simplemente se añade a la URL donde está alojada la API de Google al cargar la librería JavaScript en nuestra aplicación, sustituyendo el texto `API_KEY_GOOGLEMAPS` por la clave, como en el siguiente ejemplo:

```
<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?key=API_KEY_GOOGLEMAPS&sensor=true">
</script>
```

Ilustración 21 - Código HTML para cargar la librería de JavaScript de Google Maps

Una vez cargada la librería en el navegador del cliente, para insertar un mapa es necesario añadir un elemento DIV, al que se le pueden definir tamaño, posición, etcétera, y sobre el que se va a cargar el mapa de Google. El siguiente código de ejemplo crea una capa llamada `map_canvas`, que ocupará toda la ventana del navegador, y define una función llamada `initialize()`, que se ejecutará cuando se cargue la página web al completo, que utilizará la capa `map_canvas` para cargar un mapa de Google, centrado en las coordenadas -22.910, -43.202 (Río de Janeiro), con un nivel de zoom 8, mostrando la vista de mapa de carretera:

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?key=API_KEY_GOOGLEMAPS&sensor=true">
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(-22.910, -43.202),
          zoom: 8,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),
          mapOptions);
      }
    </script>
  </head>
  <body onload="initialize()">
    <div id="map_canvas" style="width:100%; height:100%"></div>
  </body>
</html>

```

Ilustración 22 - Código de ejemplo para cargar un mapa que muestra como centro la ciudad de Río de Janeiro



Ilustración 23 - Mapa de Google Maps con la ciudad de Río de Janeiro como centro

Además de cargar los mapas de Google, la herramienta necesita mostrar las posiciones de los usuarios de la aplicación en el mapa. La posición de un usuario se mostrará como un marcador de color rojo en el mapa. Para insertar un marcador nuevo en un mapa se puede hacer añadiendo el siguiente código (sombreado en amarillo):

GLOSARIO

```
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?sensor=true">
    </script>
    <script type="text/javascript">
      function initialize() {
        var mapOptions = {
          center: new google.maps.LatLng(-22.910, -43.202),
          zoom: 12,
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map_canvas"),
          mapOptions);
        var marker = new google.maps.Marker({
          position: map.getCenter(),
          map: map,
          title: 'Rio de Janeiro'
        });
      }
    </script>
  </head>
  <body onload="initialize()">
    <div id="map_canvas" style="width:100%; height:100%"></div>
  </body>
</html>
```

Ilustración 24 - Código de ejemplo para añadir un nuevo marcador



Ilustración 25 - Mapa de Río de Janeiro con un marcador

GLOSARIO

El uso de JavaScript en el navegador está orientado a eventos, lo que significa que JavaScript responde a las interacciones generando eventos y espera que un programa detecte los eventos interesantes. El modelo de evento de la versión 3 del API de Google Maps distingue dos tipos de eventos:

- Los eventos de usuario (como, por ejemplo, los eventos de ratón "click") se propagan desde el DOM hasta el API de Google Maps. Estos eventos son distintos e independientes de los eventos DOM estándar.
- Las notificaciones de cambio de estado de MVC (Modelo Vista Controlador) reflejan los cambios en los objetos del API de Google Maps y se denominan mediante una convención `property_changed`.

Cada objeto del API de Google Maps exporta una determinada cantidad de eventos con nombres. Los programas interesados en determinados eventos registran detectores de eventos de JavaScript para estos eventos y ejecutan código al recibirlos mediante el registro de controladores de eventos `addListener()` en el espacio de nombre `google.maps.event`.

La herramienta desarrollada, necesita hacer uso de esta funcionalidad de la API de Google Maps, para detectar clicks sobre los marcadores de los usuarios, un cambio en la posición geográfica del usuario, y los eventos de arrastrar y soltar un marcador.

Al hacer click sobre un marcador, la herramienta muestra información sobre la posición y sobre el usuario al que pertenece el marcador. Así que además de definir un marcador en una posición concreta, la API de Google Maps permite añadir información adicional, y un controlador de evento, que esté a la espera de que el usuario haga click sobre el marcador. El siguiente ejemplo muestra cómo realizarlo (sombreado en naranja):


```

var map = new google.maps.Map(document.getElementById("map_canvas"),
    mapOptions);
var marker = new google.maps.Marker({
    position: map.getCenter(),
    map: map,
    title: 'Rio de Janeiro'
});
var myHTML = '<table><tbody><tr><td valign="top"></td><td valign="top"><table
align="left"><tbody><tr><td><b>Usuario De Prueba Sánchez</b>&nbsp;<span
id="valor_comentario_posicion" style="color:#9b9b9b;font-size:8pt">Decidí vivir
y aquí aparecí :)</span></td></tr><tr><td><span
id="valor_direccion_posicion">Bartolomeu de Gusmão Airport - Cidade Nova, Rio de
Janeiro - Estado de Rio de Janeiro, Brasil</span></td></tr><tr><td><span
style="color:#9b9b9b;font-size:8pt" id="fechaPosicionGloboUsuario">Hace 3
horas</span><span id="EditarComentario"
style="visibility:visible;">&nbsp;&nbsp;&nbsp;<small>Editar</small>&nbsp;&nbsp;&nbsp;<small>
Más posiciones</small></span></td></tr></tbody></table></td></tr></tbody></table>'
infoWindow = new google.maps.InfoWindow();
infoWindow.setContent(myHTML);

google.maps.event.addListener(marker, 'click', function()
{
    infoWindow.open(map, marker);
});
}
</script>

```

Ilustración 26 - Código de ejemplo para añadir contenido al marcador y un controlador de evento click que abra el contenido del marcador al hacer click sobre él.



Ilustración 27 - Al hacer click sobre el marcador se abre un globo con la información definida

La herramienta utiliza el API de Google Maps para obtener direcciones (nombres de calle, ciudades, etcétera) y posiciones geográficas (Latitud-Longitud).

La codificación geográfica es el proceso de transformar direcciones (como "1600 Amphitheatre Parkway, Mountain View, CA") en coordenadas geográficas (como 37.423021 de latitud y -122.083739 de longitud), que se pueden utilizar para colocar marcadores o situar el mapa.

El API de Google Maps proporciona una clase geocoder que permite codificar geográficamente las direcciones de forma dinámica a partir de los datos introducidos por el usuario. En nuestra herramienta, el usuario puede mediante un cuadro de diálogo introducir una dirección, que se muestra en el mapa en forma de marcador, con la información de la dirección. El siguiente ejemplo muestra cómo definir un objeto de la clase geocoder, y a través de una dirección textual, cómo obtener una posición en formato Latitud-Longitud para mostrar el resultado en el mapa:

```
<script type="text/javascript">
var geocoder;
var map;
function initialize() {
    geocoder = new google.maps.Geocoder();
    var latlng = new google.maps.LatLng(-34.397, 150.644);
    var mapOptions = {
        zoom: 12,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
    map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
}
function codeAddress() {
    var address = document.getElementById("address").value;
    geocoder.geocode( { 'address': address}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            map.setCenter(results[0].geometry.location);
            var marker = new google.maps.Marker({
                map: map, position: results[0].geometry.location
            });
        }
    });
}
</script>
</head>
<body onload="initialize()">
<div id="map_canvas" style="width: 320px; height: 480px;"></div>
<div>
    <input id="address" type="text" value="Rio de Janeiro">
    <input type="button" value="Buscar!" onclick="codeAddress()">
</div>
</body>
```

Ilustración 28 - Código de ejemplo de búsqueda de direcciones



Ilustración 29 - Resultado de una búsqueda de una dirección y resultado en el mapa en forma de marcador

Mediante el uso de la posición geográfica que el navegador obtiene a través del uso del geoposicionamiento en formato Latitud-Longitud, el API de Google Maps es capaz de realizar una codificación inversa y relacionar una ubicación en el mapa con una dirección interpretable por humanos.

La clase geocoder además permite realizar directamente el proceso de codificación geográfica inversa. En lugar de transmitir una dirección (address) textual, hay que especificar un par latitud/longitud separado por comas en el parámetro latLng. El siguiente ejemplo muestra cómo el Objeto de la clase geocoder obtiene a partir de unas coordenadas en formato google.maps.LatLng una dirección textual de la ubicación en el mapa:

```

    map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
}
function inverseCodeAddress() {
    var address = document.getElementById("address").value;
    var latlngStr = address.split(", ", 2);
    var lat = parseFloat(latlngStr[0]);
    var lng = parseFloat(latlngStr[1]);
    var latlng = new google.maps.LatLng(lat, lng);
    geocoder.geocode({'latLng': latlng}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            if (results[1]) {
                map.setZoom(11);
                marker = new google.maps.Marker({
                    position: latlng,
                    map: map
                });
                infowindow.setContent(results[1].formatted_address);
                infowindow.open(map, marker);
            }
        } else {
            alert("Geocoder failed due to: " + status);
        }
    });
}
</script>
</head>
<body onload="initialize()">
<div id="map_canvas" style="width: 320px; height: 480px;"></div>
<div>
    <input id="address" type="text" value="-22.92,-43.20">
    <input type="button" value="Buscar!" onclick="inverseCodeAddress()">
</div>
</body>

```

Ilustración 30 - Código de ejemplo para realizar una búsqueda inversa



Ilustración 31 - Resultado de una búsqueda por latitud-longitud (búsqueda inversa)

4.1.5.2 Implementación del código

Como se ha comentado anteriormente, el proyecto se ha realizado mezclando varios lenguajes: HTML, JavaScript y CSS para la presentación de los datos, PHP, y SQL para el procesado y almacenamiento de los datos, con la ayuda de la técnica de desarrollo web AJAX; además de las librerías externas explicadas en el apartado anterior.

La estructura de ficheros de la herramienta desarrollada es la siguiente:

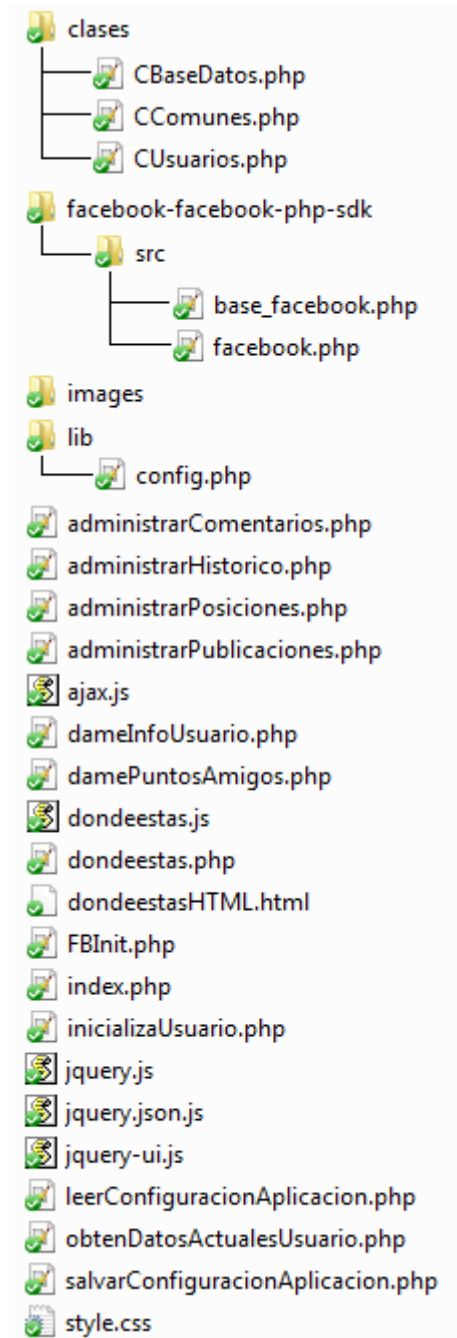


Ilustración 32 - Estructura de ficheros del proyecto

El directorio raíz del servidor Web contendrá todos los ficheros y carpetas de la ilustración anterior. Se deberá configurar el servidor web para que la página de inicio de la herramienta web sea index.php.

4.1.5.2.1 Carpeta clases

La carpeta clases, contiene las clases definidas para el proyecto. Estas clases son tres:

- **CBaseDatos:** Contiene los métodos necesarios para conectar con la base de datos, realizar consultas y ejecutar comandos de inserción, actualización y borrado de datos.

- CUsuarios: Contiene definida toda la funcionalidad para un usuario registrado: gestión de posiciones, gestión de comentarios, gestión de la aplicación, etcétera.
- CComunes: Está pensada para contener los métodos no incluidos en las otras dos clases. Actualmente contiene un método que calcula el tiempo que ha pasado desde una fecha concreta en formato “Hace X segundos/minutos/horas/días/años”

La Clase CBaseDatos está compuesta por 5 métodos:

```
function conectar()  
{  
    if(!($con=@mysql_connect($GLOBALS['db_ip'],$GLOBALS['db_user'],$GLOBALS['db_pass'])))  
    {  
        echo"Error al conectar a la base de datos";  
        exit();  
    }  
    if (!@mysql_select_db($GLOBALS['db_name'],$con))  
    {  
        echo "Error al seleccionar la base de datos";  
        exit();  
    }  
    return true;  
}
```

Ilustración 33 - Método conectar() de la clase CBaseDatos

El método conectar() realiza una conexión a la base de datos usando las credenciales definidas en el fichero lib/config.php. En el caso de que la conexión sea correcta se devuelve true.

```
function quote($valor)  
{  
    if((is_string($valor)) && ($valor != null) && ($valor != ""))  
    {  
        $valor = str_replace("\'", "&quot;", $valor);  
        $valor = str_replace('"', "'", $valor);  
        $valor = htmlentities($valor, ENT_QUOTES, 'UTF-8');  
    }  
    return $valor;  
}
```

Ilustración 34 - Método quote de la clase CBaseDatos

El método quote(\$valor) recibe por parámetros una cadena de texto que formatea, convirtiendo todos los caracteres en formato HTML (por ejemplo, el carácter “<” se convertirá en el texto “<”) y devolviéndolos formateados. Este método es útil a la hora de componer sentencias SQL que impliquen datos que puedan poner en peligro la integridad de los datos alojados en la base de datos, y también evitar el uso de códigos maliciosos en formato HTML.


```
function obtenerFila($sql)
{
    //usamos el metodo conectar para realizar la conexion
    if($this->conectar()==true)
    {
        $result = @mysql_query($sql);
        $row = mysql_fetch_assoc($result);
        return $row;
    }
    return null;
}
```

Ilustración 35 - Método obtenerFila() de la clase CBaseDatos

El método obtenerFila(\$sql) ejecuta en la base de datos el valor contenido en la variable \$sql que recibe por parámetros, para devolver el resultado de una única fila. Este método es útil cuando realizamos consultas sobre base de datos que sabemos que sólo devolverán una fila, como los datos de un usuario concreto, o la configuración de la aplicación.

```
function obtenerFilas($sql)
{
    //usamos el metodo conectar para realizar la conexion
    if($this->conectar()==true)
    {
        $result = @mysql_query($sql);
        return $result;
    }
    return null;
}
```

Ilustración 36 - Método obtenerFilas() de la clase CBaseDatos

El método obtenerFilas(\$sql) ejecuta en la base de datos el valor contenido en la variable \$sql que recibe por parámetros, y devuelve el resultado completo de esa consulta en un objeto. Este método es útil cuando realizamos consultas sobre base de datos que van a dar más de un resultado, como por ejemplo el histórico de posiciones de un usuario, o la información de las posiciones de los amigos del usuario. Desde el lugar del código desde donde se invoque este método, se deberá recorrer el resultado obtenido para leer todas las filas que contiene el objeto con la información solicitada.

```
function ejecutar($sql)
{
    if($this->conectar()==true)
    {
        $result = mysql_query($sql);
        return true;
    }
    return false;
}
```

Ilustración 37 - Método ejecutar() de la clase CBaseDatos

El método ejecutar(\$sql) ejecuta en la base de datos el valor contenido en la variable \$sql que recibe por parámetros. Se utiliza este método para ejecutar, sobre la base de datos, comandos de inserción, borrado y actualización de datos.

La clase CUsuarios contiene numerosos métodos, además de dos variables empaquetadas dentro de la clase. Se define también un método con el mismo nombre de la clase, de tal manera que cuando se instancia un nuevo objeto de la clase CUsuarios, se realizan las acciones definidas en el método CUsuarios():

```
class CUsuario
{
    var $db = null;
    var $facebook = null;

    function CUsuario()
    {
        //Salvamos el objeto de la conexion a la BD ya que se utiliza a menudo
        $db = new CBaseDatos;
        $this->db = $db;

        //Salvamos el objeto facebook
        $fb = new Facebook(array(
            'appId' => $GLOBALS['app_id'],
            'secret' => $GLOBALS['api_secret']
        ));

        $this->facebook = $fb;
    }
}
```

Ilustración 38 - Extracto de la definición de la clase CUsuarios

Como se puede observar, se definen dos variables en el contexto de la clase. La variable \$db, contendrá una instancia de la clase CBaseDatos, por lo que nos permitirá utilizar todos los métodos anteriormente descritos para la clase. La variable \$facebook, contendrá una instancia de la clase Facebook, lo que nos permitirá utilizar toda su funcionalidad, y realizar llamadas a el API de Facebook.

De los numerosos métodos definidos en la clase CUsuarios, se ilustran a modo de ejemplo algunos de ellos:

```
function crearUsuario($IDUsuario, $PermisoGeolocalizacion, $PermisoMuroFB, $CorreoPersonal)
{
    $IDUsuario = $this->db->quote($IDUsuario);
    $PermisoGeolocalizacion = $this->db->quote($PermisoGeolocalizacion);
    $PermisoMuroFB = $this->db->quote($PermisoMuroFB);
    $CorreoPersonal = $this->db->quote($CorreoPersonal);

    $sql = "INSERT INTO `FB_Usuarios` (`IDUsuario`, `FechaRegistro`, `ZoomMapa`,
    `Muro`, `Geolocalizacion`, `CorreoPersonal`)
    VALUES (". $IDUsuario .", ". time() .", 12, ". $PermisoMuroFB .", ". $PermisoGeolocalizacion .
    ", ". $CorreoPersonal .")";
    return $this->db->ejecutar($sql);
}
```

Ilustración 39 - Método crearUsuario() de la clase CUsuarios

Una vez se ha instanciado un objeto de la clase CUsuario, la variable \$this->db contiene un objeto instanciado de la clase CBaseDatos con todos sus métodos. El método crearUsuario compone una sentencia SQL de inserción con los datos recibidos por

parámetros para un nuevo usuario. Previamente, ha comprobado en la base de datos que el usuario no existe.

```
function dameInfoUsuario($IDUsuario)
{
    $IDUsuario = $this->db->quote($IDUsuario);
    $sql = "SELECT * FROM (
        SELECT IDUsuario, Latitud, Longitud, Fecha, Comentario, IDPosicion
        FROM FB_PuntosUsuario
        ORDER BY Fecha desc) A
    WHERE IDUsuario = ".$IDUsuario."
    LIMIT 1";
    return $this->db->obtenerFilas($sql);
}
```

Ilustración 40 -Método dameInfoUsuario() de la clase CUsuarios

En este caso se ilustra cómo se realiza una consulta a base de datos sobre los datos de un usuario concreto, devolviendo al final una única fila con la información del usuario.

```
function dameHistoricoUsuario($IDUsuario)
{
    $IDUsuario = $this->db->quote($IDUsuario);
    $sql = "SELECT * FROM `FB_PuntosUsuario` WHERE `IDUsuario` = '".$IDUsuario."'
        ORDER BY `Fecha` DESC
        LIMIT 15";

    $result = $this->db->obtenerFilas($sql);
    if (!$result)
    {
        return null;
    }
    else
    {
        return $result;
    }
}
```

Ilustración 41 - Método dameHistoricoUsuario() de la clase CUsuarios

En este caso, se realiza la consulta para obtener más de una fila como resultado, por lo que se utiliza el método obtenerFilas() de la clase CBaseDatos.

```

function administrarPosiciones($idsAmigos)
{
    $sql = "SELECT * FROM (SELECT IDUsuario, Latitud, Longitud, Fecha, Comentario, IDPosicion
        FROM FB_PuntosUsuario order by IDUsuario, Fecha desc) A WHERE IDUsuario IN (".$idsAmigos.")
        GROUP BY IDUsuario";
    $result = $this->db->obtenerFilas($sql);
    if (!$result) {
        $res[0]["Error"] = str_replace("\'", """, mysql_error());
    }
    else {
        $i=0;
        while ($row = mysql_fetch_assoc($result)) {
            $ObjTime = new CComunes;
            $diferenciaTiempo = $ObjTime->nicetime($row['Fecha']);
            $r = $this->facebook->api('/'. $row['IDUsuario']);
            if ($r)
            {
                $res[$i]["Nombre"] = $r['name'];
                $res[$i]["Pic"] = "https://graph.facebook.com/".$r['id']."/picture";
                $res[$i]["IDUsuario"] = $row['IDUsuario'];
                $res[$i]["Latitud"] = $row['Latitud'];
                $res[$i]["Longitud"] = $row['Longitud'];
                $res[$i]["Fecha"] = $diferenciaTiempo;
                $res[$i]["Comentario"] = str_replace("\'", """, html_entity_decode($row['Comentario']));
                $res[$i]["IDComentario"] = $row['IDPosicion'];
                $res[$i]["URL"] = $r['link'];
                $res[$i]["ID"] = $i;
            }
            else {
                $res[0]["Error"] = "Error al conectar con la BD de FB";
                return $res;
            }
            $i++;
        }
        return $res;
    }
}

```

Ilustración 42 - Método administrarPosiciones() de la clase CUsuarios

Este último método de ejemplo, ilustra cómo hacer uso de la clase CComunes, para obtener de manera textual, el tiempo que ha pasado desde que se almacenó una posición del usuario; cómo utilizar el objeto facebook almacenado para obtener información de un usuario a través del API, y cómo se almacenan los datos de un usuario en un array multidimensional, que finalmente se enviará en formato json como se explicará más adelante.

Para finalizar, la clase CComunes contiene un único método, nicetime(\$date), que convierte una fecha en formato UNIX, comparándola con la fecha actual, en una fecha del tipo “Hace X segundos, minutos, horas, días, años, décadas”, donde X es un entero. Este texto se utiliza para añadir información en las posiciones y los comentarios de las posiciones, sobre el tiempo que hace que se almacenó la posición o el comentario.

4.1.5.2.2 Carpeta facebook-facebook-php-sdk

Contiene los archivos fuente del SDK de Facebook, con las clases base_facebook.php y facebook.php explicadas anteriormente.

4.1.5.2.3 Carpeta images

Carpeta que contiene las imágenes que se utilizan en la herramienta.

4.1.5.2.4 Carpeta lib

En la carpeta lib, se encuentra el fichero config.php, donde se definen de manera global las constantes y sus valores, que van a ser utilizadas por la herramienta para conectar con la base de datos, para conectar con Facebook, y la clave de API de Google Maps.

```

$api_key      = 'lashdfowenvfilwejklsd3powjkldfjk';
$api_secret   = 'alksdfowerjwerjl23piopi24piwepio';
$api_key_googlemaps = 'sakdf23jsdfjkjvjknm,sdfm,o123okwekhkwek';
$app_id       = '898887615443';
$base_fb_url  = 'connect.facebook.com';

$db_host      = 'localhost';
$db_user      = 'db_user';
$db_pass      = 'TTsdfq3SDfvse';
$db_name      = 'dondeestas';

```

Ilustración 43 - Constantes globales definidas en el fichero config.php

4.1.5.2.5 Página de inicio de la herramienta

El fichero index.php es el primer archivo que se ejecutará en el servidor Web cuando un usuario le haga una petición HTTP. El fichero index.php sirve básicamente para cargar el fichero FBInit.php y el fichero dondeestas.php, además de recoger excepciones de ejecución y la variable POST Sesion, que se utilizará para hacer logout de la aplicación.

El fichero FBInit.php hace uso del SDK de Facebook para crear un objeto de la clase Facebook, y comprobar si el usuario tiene en el servidor una sesión activa, que permita leer y escribir datos en la red social Facebook. Ese objeto se almacena en una variable de sesión llamada \$facebook, y servirá al fichero dondeestas.php para decidir si cargar la página de inicio de sesión (dondeestasHTML.html), o el contenido de la página principal de la herramienta.

La página de inicio de sesión (dondeestasHTML.html) es una página en lenguaje HTML, que contiene una breve explicación de la herramienta, y un botón de inicio de sesión en la misma. En esta página se carga el SDK de Facebook para JavaScript, mediante el cual es usuario se autenticará en Facebook, y permitirá a nuestra herramienta el acceso a sus datos.

Una vez el usuario se ha autenticado y ha permitido el acceso a sus datos a la herramienta, en el sistema existirá una variable de sesión del usuario de Facebook válida y la página dondeestas.php cargará el contenido de la página de inicio de la herramienta.

```

<?
include_once 'lib/config.php';
include_once('facebook-facebook-php-sdk/src/facebook.php');

global $facebook, $user, $user_profile, $logoutUrl, $loginUrl;

$facebook = new Facebook(array(
    'appId' => $GLOBALS['app_id'],
    'secret' => $GLOBALS['api_secret']
));

//En $user tenemos el ID del usuario que ha hecho login
$user = $facebook->getUser();

//Comprobamos si el usuario que ha hecho login tiene un token de acceso para la aplicacion
if ($user) {
    try {
        //Para comprobar si el login es correcto en la aplicación, intentamos realizar
        //una operacion que solo esta permitida si se ha dado acceso a la aplicación
        $user_profile = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        //Se ha hecho login en Facebook, pero no en la aplicación
        error_log($e);
        $user = null;
    }
}

if ($user) {
    $logoutUrl = $facebook->getLogoutUrl();
} else {
    $loginUrl = $facebook->getLoginUrl();
}
?>

```

Ilustración 44 - Código fuente del fichero FBInit.php

Desde el fichero dondeestas.php se realiza una llamada al script inicializaUsuario.php, que comprueba si el usuario existe o no en el sistema. En el caso de que no exista, el usuario se da de alta. Además, se cargan las librerías JavaScript ajax.js, dondeestas.js, jquery.js, jquery.json.js y jquery-ui.js, y el archivo de hoja de estilos CSS style.css.

Los tres ficheros, jquery.js, jquery.json.js y jquery-ui.js, componen la librería de JQuery. JQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

El script dondeestas.js es un fichero escrito en lenguaje JavaScript, en el que se definen todas las funciones disponibles para manipular la herramienta del lado del cliente. A través de este script, se añaden las marcas de posición del usuario y sus amigos en el mapa, los globos de información de las posiciones, se obtiene y muestra la dirección geográfica del usuario, se definen los eventos de las marcas de posición (por ejemplo, al hacer click sobre una posición, se muestra un globo de información); se muestra o se oculta la capa de comentarios, se definen las acciones de cada botón de la herramienta, se muestra la configuración de la herramienta.

El script ajax.js también es un fichero escrito en lenguaje JavaScript. En este script se definen las funciones necesarias para comunicarse con el servidor para obtener o añadir información, de manera asíncrona, en segundo plano. Para ello se utiliza la librería anteriormente mencionada, JQuery, y el objeto ajax que proporciona. Un ejemplo para ilustrar el funcionamiento de llamada asíncrona usando AJAX es el siguiente:

```
function mostrarComentarioPosicionAmigos(IDUsuarioPosicion, IDPosicion)
{
    var divCapaComentariosMenu = document.getElementById("divShowComentarioPosicionAmigos");
    $.ajax(
    {
        url:         'administrarComentarios.php',
        type:        'POST',
        dataType:    'json',
        data:        "IDUsuarioPosicion=" + IDUsuarioPosicion + "&IDPosicion=" + IDPosicion + "&Tipo=2",
        beforeSend:  function() {
            divCapaComentariosMenu.innerHTML = "<img src='images/espera-cargando.gif' border='0' />" +
            "<small> Cargando comentarios...</small>";
        },
        error:       function() {
            alert('Se ha producido un error al cargar los datos de la aplicación. ' +
            'Por favor, recargar la página de nuevo');
        },
        success:     function(response){
            var obj = response;
            if(obj != null){
                var Respuesta = obj;
                if(Respuesta.datos == ""){
                    divCapaComentariosMenu.innerHTML = "Sesion caducada.";
                }
                else{
                    divCapaComentariosMenu.innerHTML = Respuesta.datos;
                }
            }
            divCapaComentariosMenu.style.visibility = "visible";
        },
        complete:    function() {}
    });
}
```

Ilustración 45 - Función que usa AJAX para carga asíncrona de datos

La función `mostrarComentariosPosicionAmigos`, se utiliza en la herramienta para mostrar en la barra lateral izquierda los comentarios en una posición de un amigo. El objeto `jQuery.ajax` se configura con una serie de parámetros, que pueden ser valores o funciones.

- `url`: cadena que contiene la dirección URL a la que se realiza la petición. En nuestro caso, el script `administrarComentarios.php`
- `type`: contiene el tipo de petición a realizar, GET o POST. Por defecto, el valor es GET. En nuestro caso, usamos el tipo POST.
- `dataType`: contiene el tipo de data que se espera recibir como respuesta del servidor. Puede ser xml, html, json, text entre otros. En nuestro caso, esperamos como respuesta el tipo JSON como objeto JavaScript.
- `data`: contiene los datos a ser enviados al servidor. En nuestro caso, para realizar la petición para obtener los datos de los comentarios de un usuario, necesitamos enviar al servidor el identificador del usuario, de la posición, e indicar que queremos leer los comentarios al enviar el valor `Tipo=2`.
- `beforeSend`: se define una función que se ejecuta antes de realizar la solicitud. En nuestro caso, sustituimos el contenido de la capa donde se muestran los comentarios por una imagen de carga de datos y el texto “Cargando comentarios...”
- `error`: se define una función que se ejecuta en el caso de que ocurra un error.
- `success`: se define una función que se ejecuta en el caso de que la solicitud haya sido correcta. El parámetro `response`, contiene un objeto del tipo que se haya definido en el parámetro `type`. En nuestro caso, contiene un objeto JavaScript en formato json, y el contenido de la respuesta se almacena en `response.datos`.

GLOSARIO

- complete: se define una función que se ejecuta una vez la solicitud haya finalizado (después de que se ejecuten las funciones error y success)

Desde el script ajax.js, se realizan las llamadas a los siguientes scripts en php:

administrarComentarios.php. Este script gestiona las peticiones de inserción, borrado, consulta total. Recibe por parámetros POST el contenido del parámetro Tipo, que puede tomar los valores 0 (marcar como borrado), 1 (insertar), 2 (consultar), 3 (consultar los 3 últimos comentarios). Además dependiendo del tipo de operación, recibe otros parámetros como el identificador del usuario, de la posición, comentario y usuario que realiza el comentario. Como respuesta, envía un objeto en formato JSON, que contiene la información con los datos solicitados.

administrarHistorico.php: Este script realiza una consulta a la base de datos, de las últimas posiciones almacenadas en el sistema. Recibe por parámetros POST el identificador del usuario del cual se quiere obtener la información. Como respuesta, envía un objeto en formato JSON con la información del histórico.

administrarPosiciones.php: Este script gestiona las peticiones de inserción y actualización de posiciones de un usuario. Recibe por parámetros POST el identificador de usuario, el identificador de posición, y el comentario sobre la posición (Qué haces aquí?). Si el identificador de posición se envía vacío, significa que se desea insertar un nuevo punto. En caso contrario, sólo se desea actualizar el contenido del comentario de la posición. Como respuesta a la solicitud, se envía un objeto JSON con la información de la posición del usuario actualizada (con el nuevo punto, o bien con el mismo punto, pero con el comentario actualizado).

administrarPublicaciones.php: Este script gestiona las publicaciones en el muro del usuario en Facebook. Recibe como parámetro POST el tipo de publicación, y la información relevante acorde al tipo de publicación. El tipo de publicación puede tomar los valores 1 (si cuando el usuario instala la aplicación la primera vez decide publicarlo), 2 (cuando publica una nueva posición) y 3 (cuando comenta una posición de un amigo). Como respuesta, envía en texto plano el identificador de la publicación en Facebook, o en el caso de error, una cadena vacía.

dameInfoUsuario.php: Este script gestiona las peticiones de consulta sobre un usuario. Obtiene el identificador del usuario de la variable de sesión PHP actual a través de la clase CUsuarios. Como respuesta, envía un objeto en formato JSON con la información actual del usuario: Nombre, URL del perfil y fotografía en Facebook, identificador de usuario, posición, comentario de la posición, fecha de la posición, y posibles errores.

damePuntosAmigos.php: Este script gestiona las peticiones de consulta sobre las posiciones de los amigos de un usuario. Obtiene el identificador del usuario de la variable de sesión PHP actual a través de la clase CUsuarios y realiza una consulta de los identificadores de los amigos en Facebook que usan la herramienta. Con esos identificadores, compone un array multidimensional con la información de cada usuario (Nombre, URL del perfil y fotografía en Facebook, identificador de usuario, posición, comentario de la posición, fecha de la posición). Como respuesta, se envía un objeto JSON con toda la información.

obtenDatosActualesUsuario.php: Este script consulta la última posición almacenada de un usuario. Recibe por parámetros POST el identificador de usuario y envía en formato JSON la información solicitada.

leerConfiguraciónAplicacion.php: Este script consulta las opciones de la herramienta definidas para un usuario. Recibe por parámetro POST el identificador de usuario, envía como respuesta en formato HTML un formulario con las opciones de la aplicación.

GLOSARIO

`salvarConfiguracionAplicacion.php`: Este script gestiona los datos de configuración de la herramienta, actualizando los datos para usuarios registrados, o dando de alta nuevos usuarios, junto con las opciones de la aplicación.

4.1.5.3 Pantalla de inicio de sesión

Contiene información acerca de la herramienta y es la página de inicio para un usuario que no se ha registrado en la aplicación.



Ilustración 46 - Página de inicio de la aplicación

Cuando el usuario hace click sobre el botón de Facebook, se abre una ventana emergente, donde se solicita acceso a su cuenta en Facebook. Esta nueva página está alojada en el servidor de Facebook, la herramienta en ningún momento tiene acceso a los datos de autenticación del usuario.



Facebook

Inicia sesión para utilizar tu cuenta de Facebook con [Dónde estás?](#)

Correo electrónico o teléfono:

Contraseña:

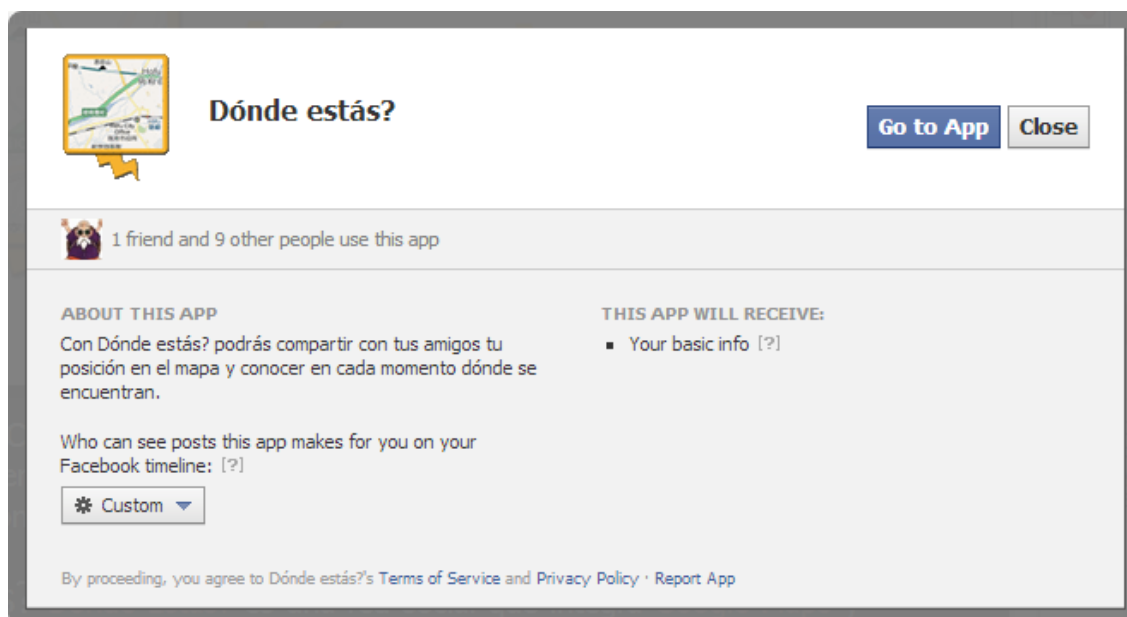
☐ No cerrar sesión

[¿Has olvidado tu contraseña?](#)

[Regístrate en Facebook](#) [Entrar](#) [Cancelar](#)

Ilustración 47 - Solicitud de acceso a Facebook para a través de la herramienta

Cuando el usuario accede a su cuenta en Facebook, y aún no ha instalado la aplicación, Facebook solicita el permiso para permitir a la aplicación el acceso a sus datos (Datos básicos del usuario, como su nombre, fotografía de perfil, URL del perfil, edad, etcétera)



Dónde estás? [Go to App](#) [Close](#)

1 friend and 9 other people use this app

ABOUT THIS APP
Con Dónde estás? podrás compartir con tus amigos tu posición en el mapa y conocer en cada momento dónde se encuentran.

Who can see posts this app makes for you on your Facebook timeline: [?]
* Custom

THIS APP WILL RECEIVE:
■ Your basic info [?]

By proceeding, you agree to Dónde estás?'s [Terms of Service](#) and [Privacy Policy](#) · [Report App](#)

Ilustración 48 - Permiso de acceso a la herramienta

Además, la herramienta necesita permisos adicionales para poder publicar contenido en el nombre del usuario, para ello, se muestra la solicitud del siguiente permiso:

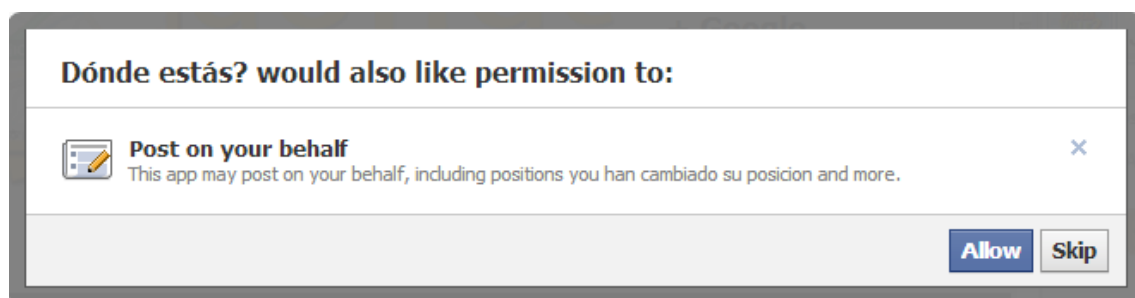


Ilustración 49 - Permiso extendido de uso de la herramienta

Una vez el usuario ha accedido a su cuenta de Facebook, y ha dado permisos a la herramienta a acceder a sus datos, la herramienta muestra un mensaje de bienvenida y de configuración de la herramienta.

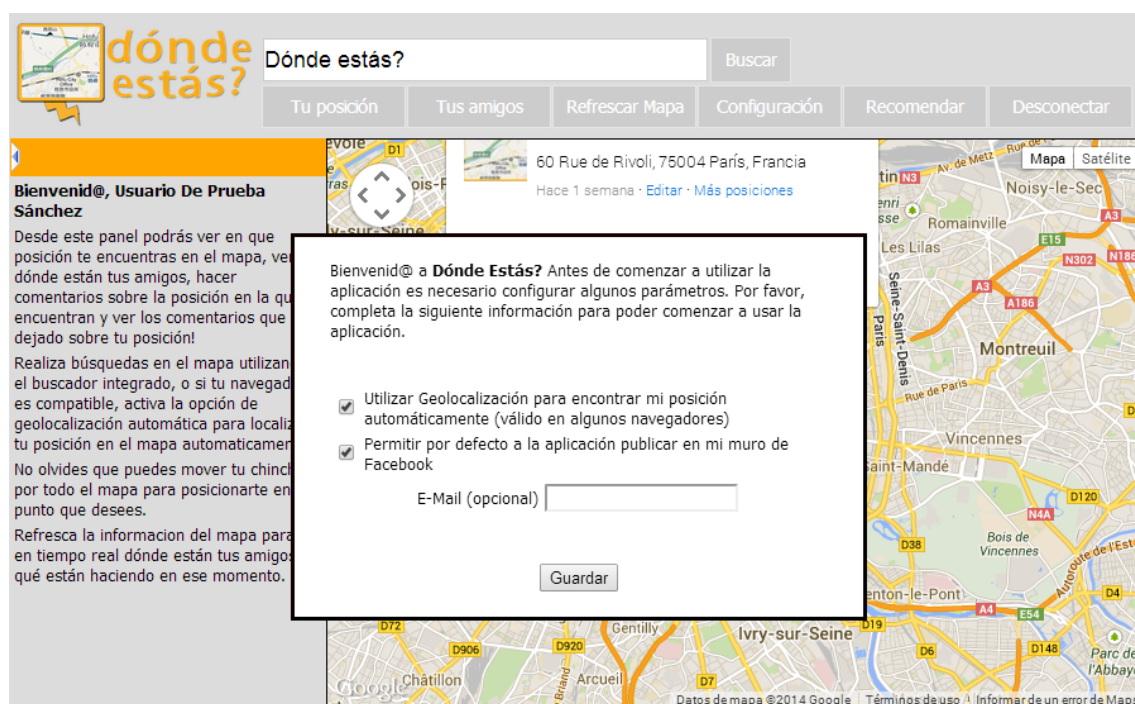


Ilustración 50 - Mensaje de bienvenida y configuración de la herramienta

Una vez se ha configurado la herramienta, la aplicación puede localizar la posición del usuario automáticamente a través del navegador (si el usuario lo ha configurado así y si el navegador es compatible).

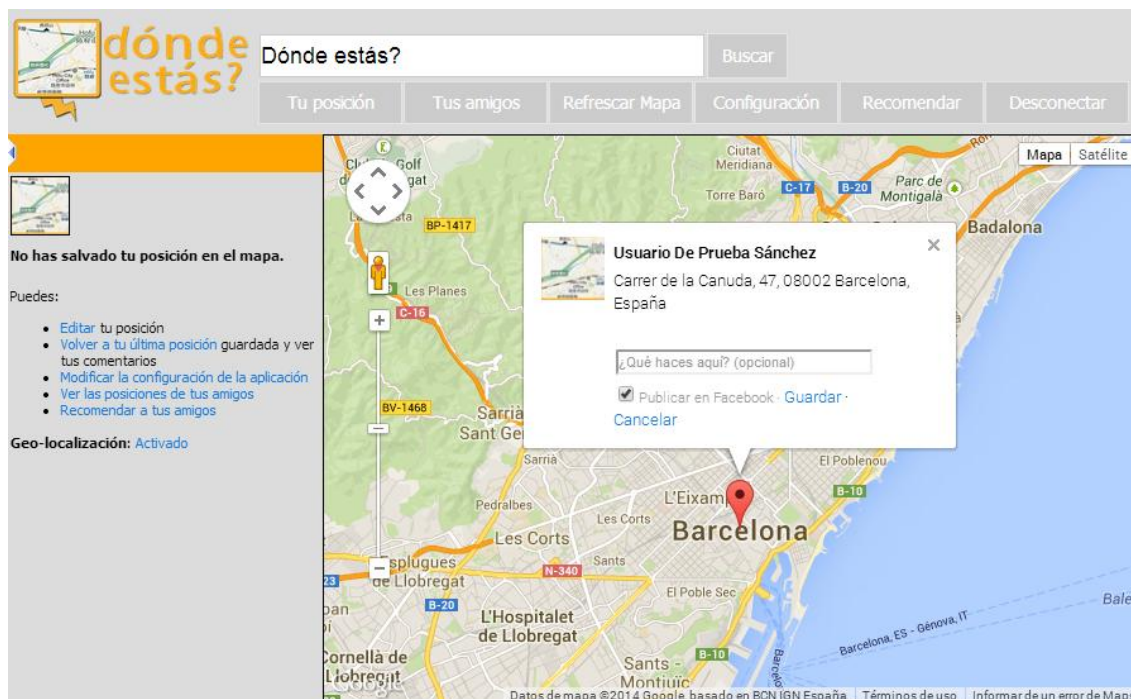


Ilustración 51 - Nueva posición en el mapa usando geoposicionamiento

Al salvar una nueva posición en el mapa, el usuario puede además publicar en su muro de Facebook esta nueva posición

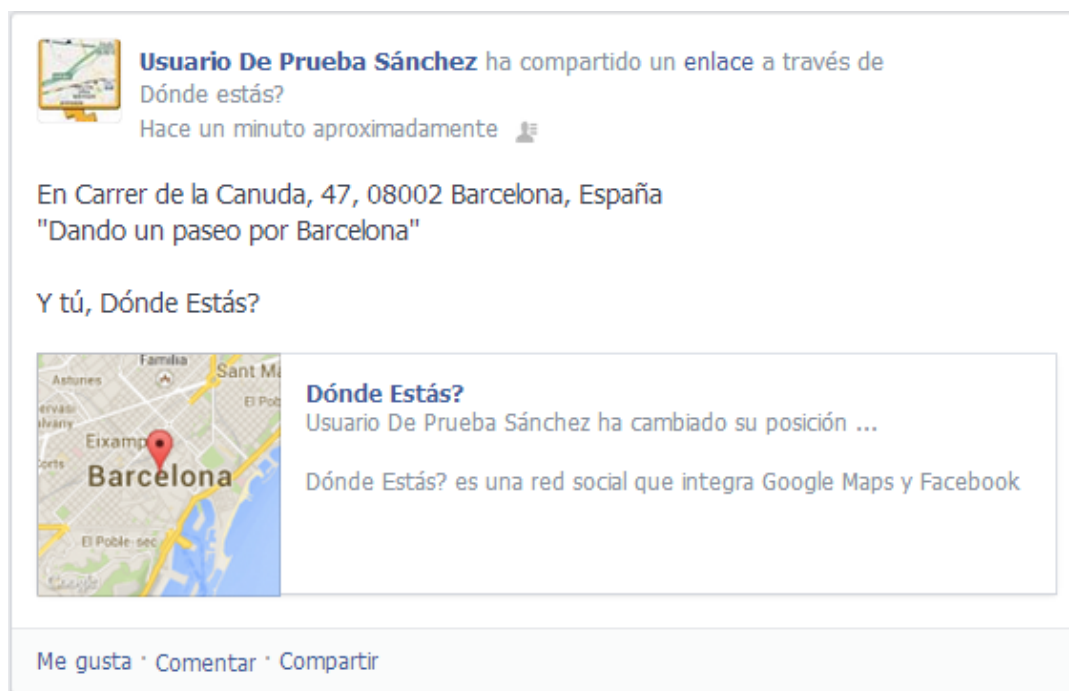


Ilustración 52 - Publicación en Facebook de nueva posición en el mapa

La herramienta también permite realizar búsquedas. Tan sólo es necesario informar el campo de búsqueda y presionar Enter o el botón Buscar. Un nuevo marcador aparecerá en el mapa, con el primer resultado de la búsqueda. La siguiente imagen muestra el resultado de buscar “Nueva York” en el mapa.

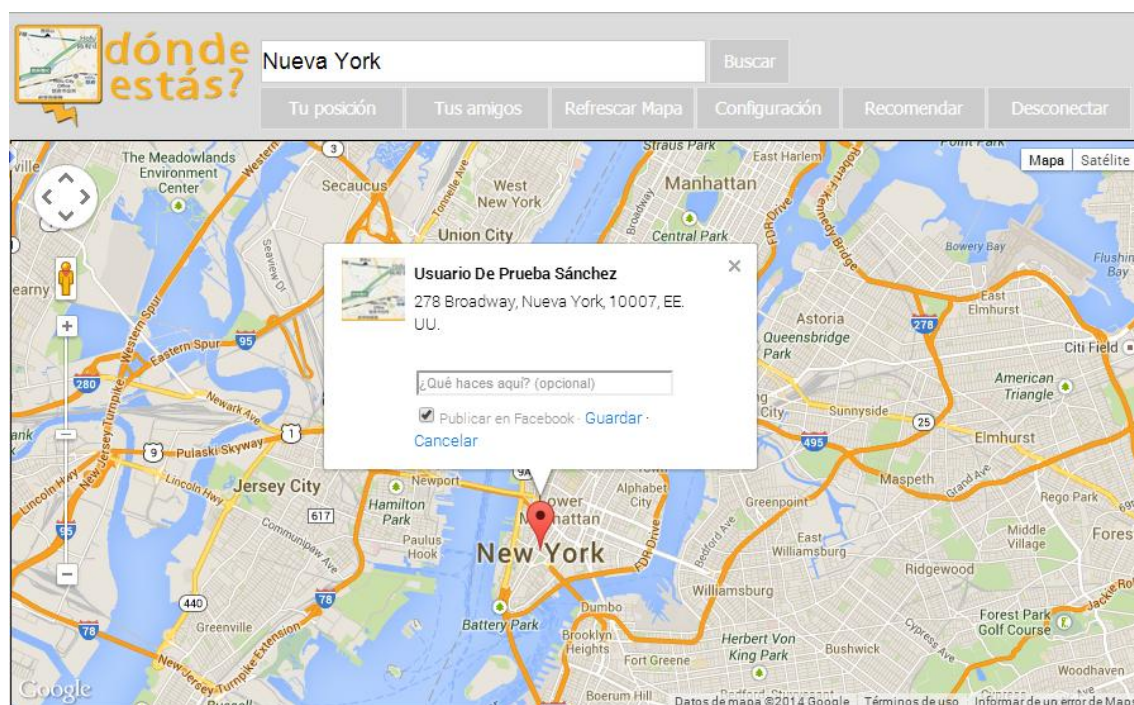


Ilustración 53 - Resultado de buscar Nueva York en el mapa

La herramienta muestra al usuario, los amigos de Facebook que utilizan la aplicación. Al hacer click sobre el botón “Tus amigos”, se muestra un listado con las imágenes de perfil de cada amigo en el panel lateral.

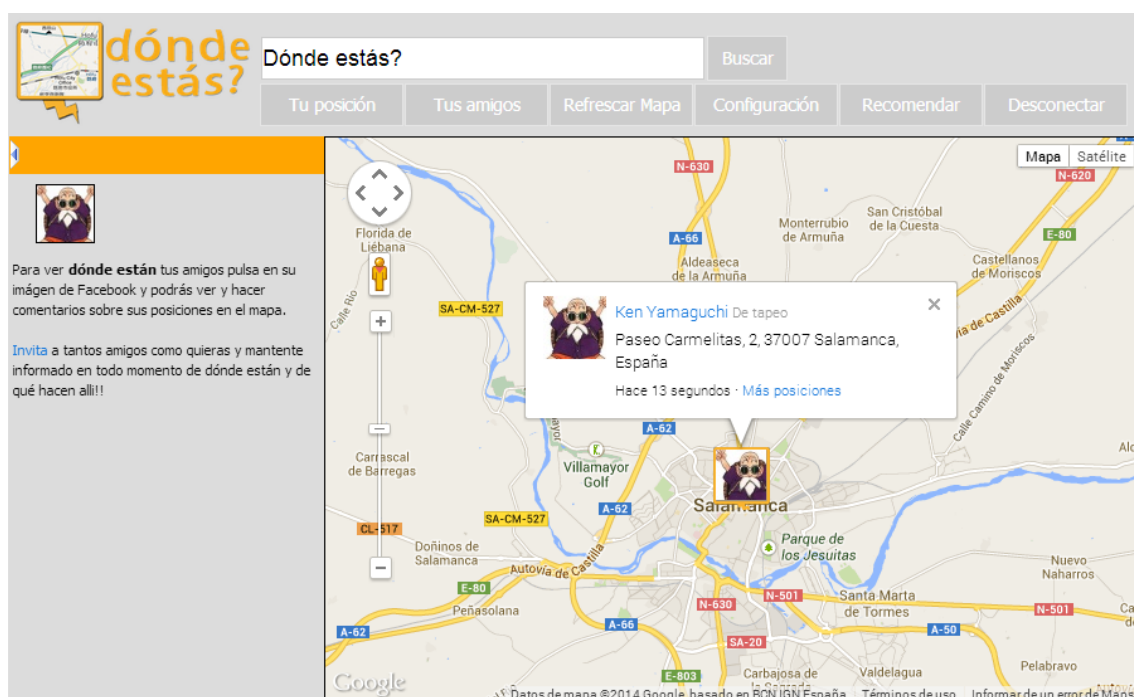


Ilustración 54 - Vista con los amigos del usuario que usan la herramienta

Al hacer click sobre la posición de un amigo, la herramienta da la posibilidad de comentar la posición del amigo.

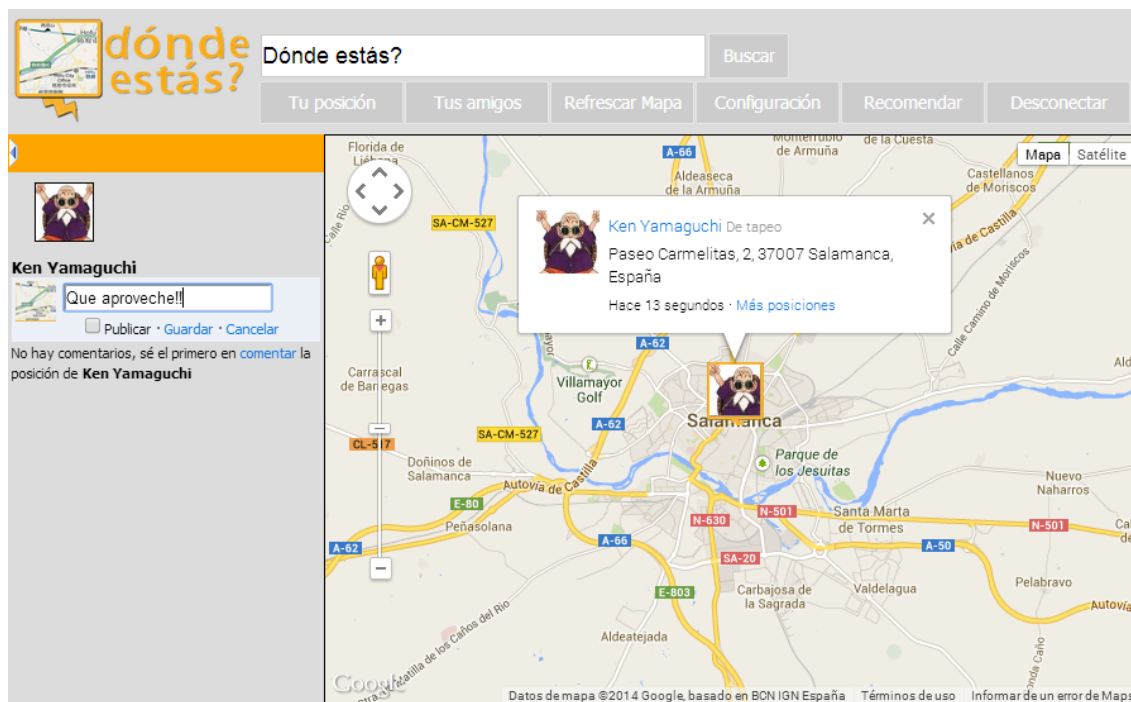


Ilustración 55 - Nuevo comentario en la posición de un amigo

Al guardar el comentario de la posición de un amigo, el sistema almacena el nuevo comentario para la posición, y muestra el resultado en el panel lateral.

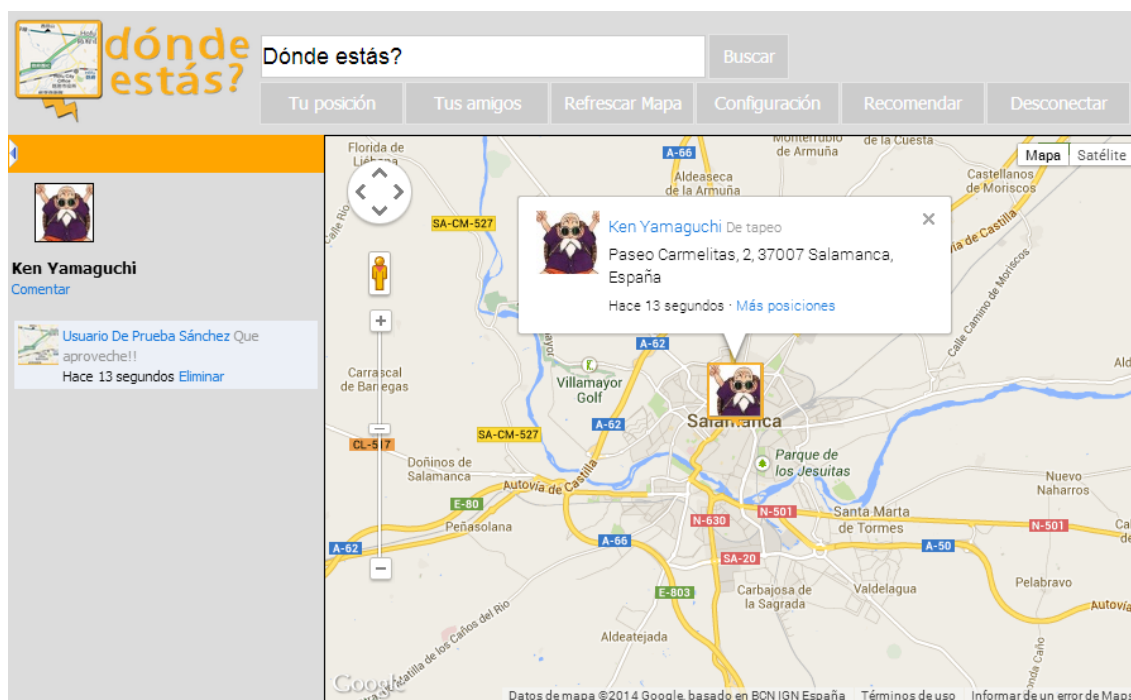


Ilustración 56 - Posición de un amigo con un comentario

La herramienta permite conocer el histórico de las posiciones de un usuario haciendo click sobre el link “Más posiciones” del globo que se muestra en la posición del usuario o de un amigo. Al hacer click, la herramienta carga en el panel lateral las últimas posiciones (hasta 15) nombradas y ordenadas alfabéticamente y los comentarios correspondientes a la posición mostrada en el mapa. En el mapa se muestra la posición

GLOSARIO

más reciente almacenada del usuario consultado, y en el globo de la posición también se muestran las flechas de navegación entre las distintas posiciones.

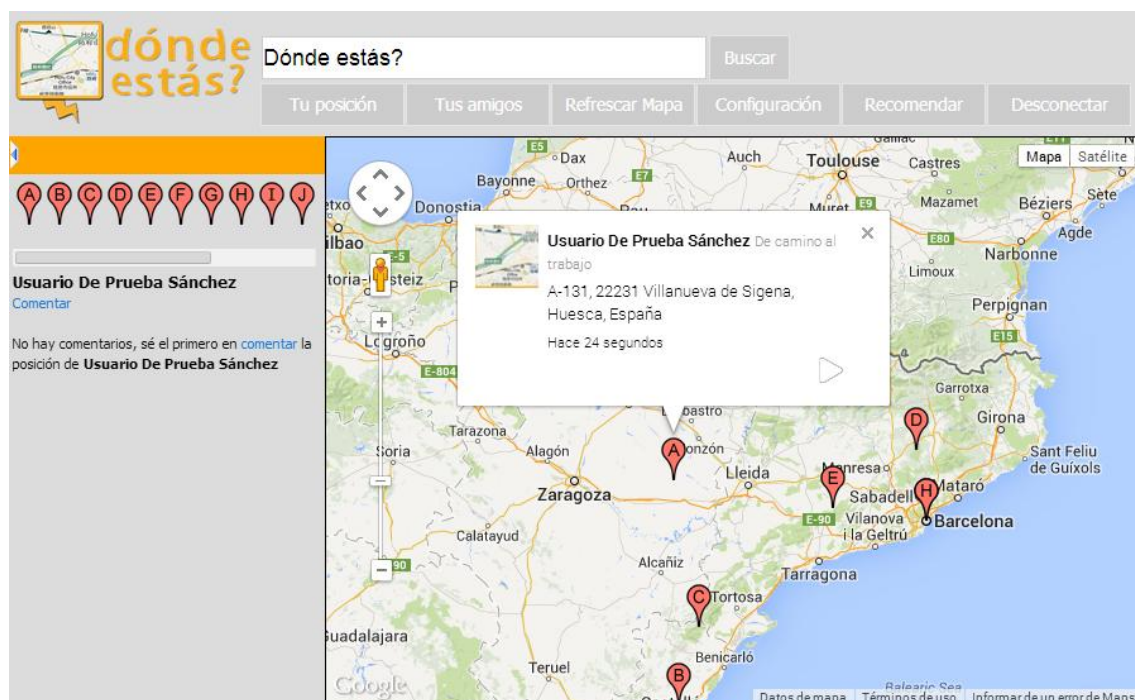


Ilustración 57 - Mapa con las últimas posiciones almacenadas del usuario

El usuario puede actualizar el contenido mostrado en el mapa haciendo click en el botón “Refrescar Mapa”, cargando las posiciones actuales de sus amigos y también la del usuario (si por ejemplo está en movimiento).

Las opciones de la aplicación se pueden modificar en cualquier momento pulsando el botón “Configuración”. El sistema permite poner por defecto el sistema de geolocalización automática (si el navegador lo soporta), y la publicación en Facebook de la actividad en la herramienta. Además permite añadir información de contacto a través del correo electrónico de manera opcional.

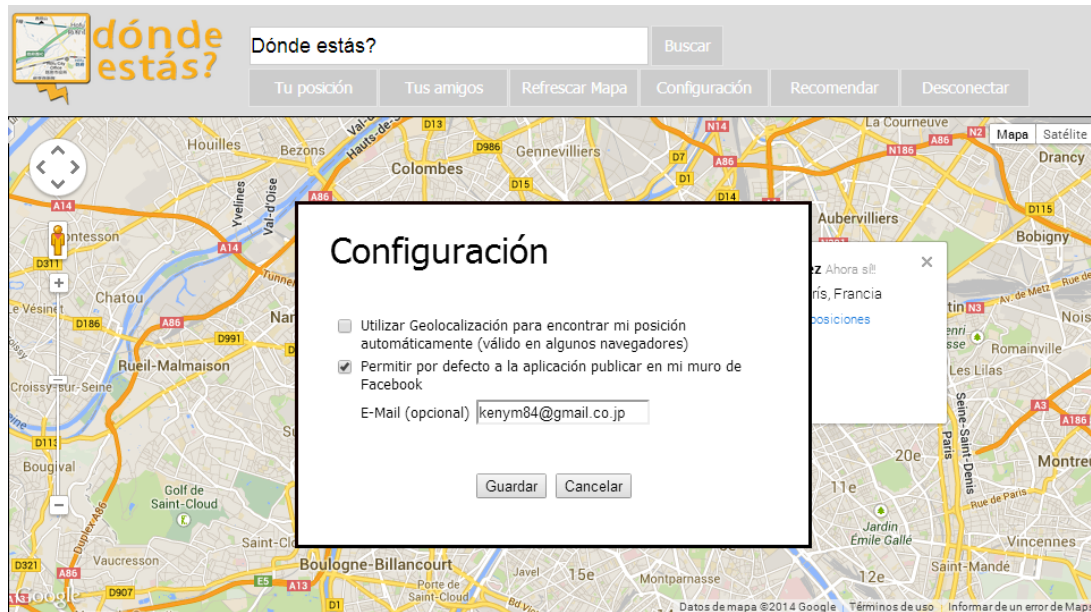


Ilustración 58 - Configuración de la aplicación

El botón “Recomendar”, permite al usuario enviar a través de Facebook invitaciones a sus amigos para usar la herramienta.

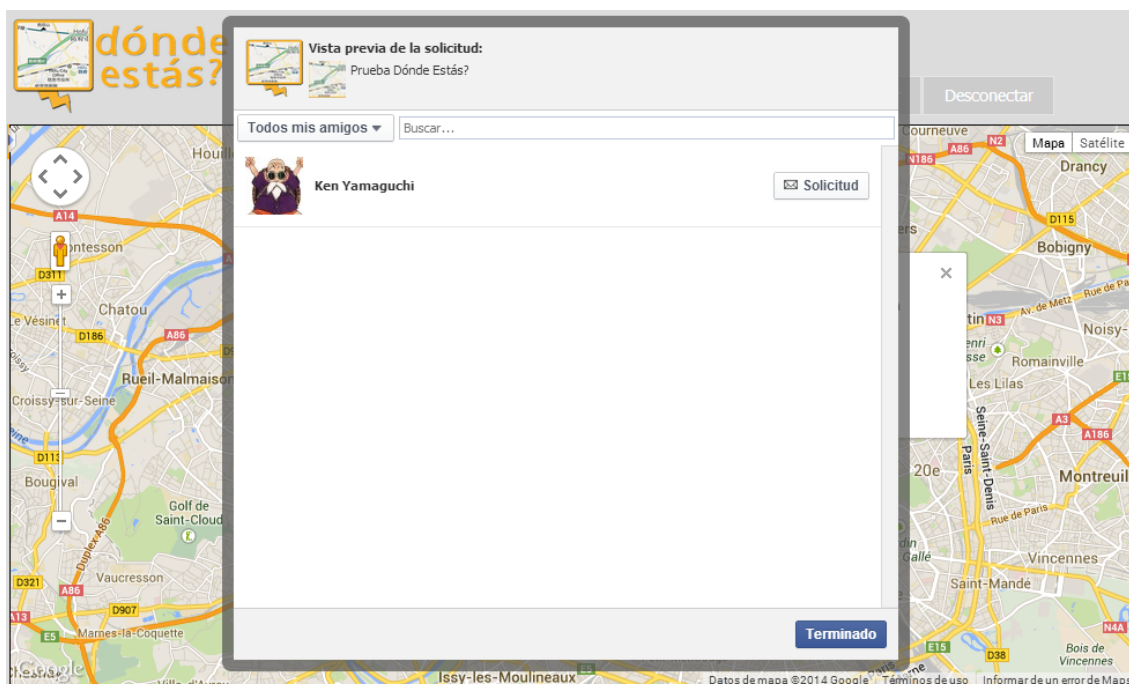


Ilustración 59 - Solicitud de envío de invitaciones de la herramienta a amigos

Por último, el botón “Desconectar” permite al usuario desconectarse de la herramienta (y también de Facebook) eliminando del sistema las variables de sesión del usuario.

Capítulo 5

Conclusiones

5.1 Conclusiones y líneas futuras

En este apartado se detallan las conclusiones obtenidas tras la finalización del desarrollo del presente proyecto. Se detalla de manera desglosada el presupuesto para el proyecto, y para finalizar se analizan los puntos menos fuertes del proyecto con el fin de mejorarlos de cara a una implementación del mismo a mayor escala.

5.1.1 Conclusiones

Con el paso del tiempo, las empresas se han percatado de la importancia de hacer un hueco en sus presupuestos para invertir en publicidad en las redes sociales, consiguiendo aumentar así su presencia en la red, llegando a un número mucho mayor de usuarios. De igual manera que a principios de siglo una empresa que no tuviera página Web, perdía credibilidad de cara a sus clientes, hoy en día ocurre lo mismo con una empresa que no tiene presencia en las redes sociales.

Es tal la importancia de las redes sociales, que éstas han hecho que cambiemos la forma en que nos comunicamos, ya que combinando con dispositivos inteligentes con acceso a Internet inmediato, nos permiten estar conectados siempre que queramos.

La posibilidad de crear una red social a través de la cual se comparen lugares en el planeta a partir del potencial de una red social consolidada como es Facebook, fue el principal motivo que me animó a empezar con este proyecto.

En el inicio del proyecto, a principios del 2009, prácticamente acababa de lanzarse públicamente en el mundo la red social Foursquare, y Facebook parecía no darse cuenta del valor añadido que suponía añadir a las publicaciones de los usuarios la posibilidad de etiquetarlas geográficamente o la idea de descubrir nuevos lugares y compartir esa información en la red social.

Se han encontrado varios problemas a la hora de implementar la herramienta. El principal ha sido el tener que aprender un nuevo lenguaje, ya que el conocimiento inicial del mismo era muy básico. La segunda dificultad, ha sido integrar las librerías tanto de Google Maps, como de Facebook, siendo esta última la más complicada. Aunque la documentación existente de ambas librerías es muy extensa y está explicada de manera clara y concisa en ambos casos, las funcionalidades más complicadas no lo están. Otra dificultad añadida ha sido depurar el código y los errores con el entorno Facebook, ya que se ha tenido que utilizar una herramienta online que proporciona la red social debido a que en local no es posible utilizar la API de Facebook (para conectar con el usuario de Facebook es necesario un Token de acceso, que sólo es válido para la dirección URL que se indica en la configuración de la aplicación). Por último, a lo largo del desarrollo, las librerías de Facebook y Google Maps también han ido evolucionando, por lo que el código de la aplicación también se ha tenido que ir adaptando a los cambios, especialmente notable ha sido el cambio en la arquitectura de la API de Facebook, que ha supuesto un añadido tanto en aprendizaje como en tener que actualizar el código con los nuevos métodos.

La planificación del proyecto, se ha ido prolongando debido a todas estas dificultades, añadidas a tener que compaginar el desarrollo del mismo con un empleo a tiempo completo. Según se avanzaba en el aprendizaje del lenguaje y en el conocimiento de las dos librerías a integrar, se han adquirido las habilidades necesarias para ser capaz de desarrollar el proyecto de manera autónoma y eficaz.

El proyecto ha pasado por todas las fases del desarrollo, partiendo desde cero en la implementación del mismo, y finalmente cumpliendo todos los objetivos marcados desde el inicio. Muchos de los conocimientos adquiridos a lo largo de la carrera se han aplicado lo que han supuesto la posibilidad de concluirlo de manera satisfactoria, cumpliendo con los objetivos planteados al principio del proyecto.

5.1.2 Líneas futuras

El sistema que se ha desarrollado, está pensado como una aplicación de escritorio, en la cual el usuario desde su ordenador de sobremesa o portátil puede interactuar con otros usuarios, descubriendo nuevo lugares y compartiendo vivencias de esos lugares que ha visitado. La tendencia actual es la movilidad, de tal manera que el usuario está siempre conectado mediante el uso de teléfonos inteligentes o tabletas. Es por eso que adaptar la aplicación a esta tendencia es uno de las metas a fijar en el futuro, bien como aplicación nativa, dependiendo del sistema operativo del teléfono, o bien como una aplicación adaptada a entornos móviles.

La idea de red social basada en la localización de los usuarios, se puede aplicar de muchas maneras, creando diversas aplicaciones con diferentes usos: desde aplicaciones que ayudan a localizar a niños y personas mayores a través de dispositivos de control de posición (GPS o los mismos teléfonos inteligentes); redes sociales gastronómicas, donde se recomiendan lugares basados en el número de visitas de los usuarios, y ciertos parámetros relacionados con sus gustos; aplicaciones que localizan parking libre usando la información que comparten otros usuarios; hasta aplicaciones para solicitar servicios como un taxi, conociendo en todo momento en qué lugar se encuentra, el tiempo estimado hasta la recogida, el coste estimado, compartiendo esta información por ejemplo con los amigos que te esperan en el lugar de destino.

Capítulo 6

Presupuesto

6.1 Presupuesto del proyecto

A continuación se detallan los costes estimados del proyecto.

6.1.1 Costes de personal

El desarrollo del proyecto se ha dividido en dos fases, fijándose el inicio de proyecto en el mes de febrero de 2009. La primera fase del desarrollo del proyecto ha sido la implementación de la aplicación (análisis, diseño e implementación), que finalizó en julio de 2010. La segunda fase, consistente en la creación de la documentación del proyecto se inició en diciembre de 2013, y ha finalizado en marzo de 2014.

El tiempo de desarrollo ha sido de 70 semanas (18 meses) con un tiempo estimado de 8 horas a la semana, por lo que el tiempo estimado de la primera fase del proyecto es de 560 horas. La segunda fase del desarrollo, la documentación, ha sido de 14 semanas, con una estimación de 5 horas semanales, lo que hace un total de 70 horas.

El tiempo total estimado de horas dedicadas al proyecto es de 630 horas, desglosadas de la siguiente manera:

GLOSARIO

Fase	Horas
Análisis	80
Diseño	100
Implementación	380
Documentación	70
Total	630

Tabla 64 - Estimación por fases

A continuación se detallan los costes teniendo en cuenta el coste por hora del personal específico para cada fase:

Fase	Coste/hora	Horas	Coste total
Análisis	32,68	80	2614.58
Diseño	32,68	100	3268.22
Implementación	20,53	380	7800.90
Documentación	20,53	70	1437.00
Total			15120.71

Tabla 65 – Coste por fases



UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

Ken Yamaguchi Miguel

2.- Departamento:

3.- Descripción del Proyecto:

- Título **Creación de una Red Social mediante Google Maps**
- Duración (meses) **21**
- Tasa de costes Indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):

Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar – solo a título informativo)	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)	Firma de conformidad
Ken Yamaguchi	23442134123-Z	Ingeniero Senior	1,371428571	4.289,54	0,00 5.882,80	
Ken Yamaguchi	23442134123-Z	Ingeniero	3,428571429	2.694,39	9.237,91 0,00 0,00	
Hombres mes 4,8				Total	15.120,71	

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Equipo de trabajo	500,00	100	21	60	175,00
Servidor Web	1.000,00	30	21	60	105,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
Total					280,00

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO^{e)}

Descripción	Empresa	Costes imputable
Total		0,00

^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	15.121
Amortización	280
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	3.080
Total	18.481

Nota: Todos los costes de este presupuesto están reflejados sin incluir el IVA (+21%)

Glosario

AJAX	<i>Asynchronous Javascript And XML</i>
API	<i>Application Programming Interface</i>
CSS	<i>Cascade Style Sheets</i>
DOM	<i>Document Object Model</i>
FQL	<i>Facebook Query Language</i>
GNOME	<i>GNU Network Object Model Environment</i>
GPL	<i>General Public License</i>
HTML	<i>HyperTextual Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
JDK	<i>Java Development Kit</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Hypertext Pre-processor</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>

GLOSARIO

Referencias

[WINDOWS] Ciclo de vida de soporte Técnico de Microsoft (Windows 7 Professional). Disponible [Internet]: <<http://support.microsoft.com/lifecycle/?p1=14481>> [Consultado Junio 2015]

[WORD] Novedades en Microsoft Office Word 2007. Disponible [Internet]: <<http://office.microsoft.com/en-us/word-help/what-s-new-in-microsoft-office-word-2007-HA010074224.aspx>> [Consultado Junio 2015]

[DIA] GNOME Wiki. Apps/Dia. Disponible [Internet]: <<https://wiki.gnome.org/Apps/Dia>> [Consultado Junio 2015]

[NETBEANS] NetBeand IDE 7.0.1 Release Information. Disponible [Internet]: <<https://netbeans.org/community/releases/70/>> [Consultado Junio 2015]

[GANTTPROJECT] GanttProject Home Page. Disponible [Internet]: <<http://www.ganttproject.biz>> [Consultado Junio 2105]

[GREENSHOT] Sourceforge. Disponible [Internet]: <<http://sourceforge.net/projects/greenshot>> [Consultado 25-11-2012]

[FACEBOOK] "Facebook: One Social Graph to Rule Them All?" CBS News. Disponible [Internet]: <<http://www.cbsnews.com/news/facebook-one-social-graph-to-rule-them-all>> [Consultado Junio 2015]

[FACEBOOK_DEVELOPERS] The Graph API. Disponible [Internet]: <<https://developers.facebook.com/docs/graph-api>> [Consultado Junio 2015]

[GOOGLEMAPS] Versión 3 del API de JavaScript de Google Maps. Disponible [Internet]: <<https://developers.google.com/maps/documentation/javascript/?hl=es>> [Consultado Junio 2015]

[APACHE] Apache HTTP Server Project. Disponible [Internet]: <http://httpd.apache.org/ABOUT_APACHE.html> [Consultado 19-12-2012]

[PHPMYADMIN] PhpMyAdmin Main Page. Disponible [Internet]: <http://www.phpmyadmin.net/home_page> [Consultado Junio 2015]

[MYSQL] MySQL Main Page. Disponible [Internet]: <<http://www.mysql.com/about/>> [Consultado Junio 2015]

[HTML] WC3 HTML. Disponible [Internet]: <<http://www.w3.org/html/>> [Consultado Junio 2015]

[CSS] WC3: Guía breve de CSS. Disponible [Internet]: <<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>> [Consultado Junio 2015]

[PHP] Página oficial de PHP. Disponible [Internet]: <<http://php.net/index.php>> [Consultado Junio 2015]

[SQL] Wikipedia. Disponible [Internet]: <<http://es.wikipedia.org/wiki/SQL>> [Consultado Junio 2015]

[INTECO] Estudio sobre la privacidad de los datos y la seguridad de la Información en las redes sociales online. Disponible [Internet] [PDF]: <https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Estudios/estudio_inteco_aped_120209_redes_sociales.pdf> [Consultado Junio 2015]

[RANDY] MONOGRÁFICO: Redes Sociales - Historia de las redes sociales. Observatorio Tecnológico, Gobierno de España. Disponible [Internet]: <<http://recursostic.educacion.es/observatorio/web/es/component/content/article/1043-redes-sociales?start=2>> [Consultado Junio 2015]

[COCKTAIL] Informe de resultados Observatorio Redes Sociales 3ª oleada. Disponible [Internet] [PDF]: <<https://s3.amazonaws.com/tca-2012/uploads/2011/02/Observatorio-RedesSociales2011.pdf>> [Consultado Junio 2015]

[GOWALLA] "Facebook buys Gowalla". CNN. Disponible [Internet]: <http://money.cnn.com/2011/12/02/technology/gowalla_facebook> [Consultado Junio 2015]

[NETCRAFT] Estadísticas históricas y de uso diario proporcionadas por Netcraft3. Disponible [Internet]: <<http://news.netcraft.com/>> [Consultado Junio 2015]

Referencias:

[BASESDATOS] El modelo de datos Entidad Relación. Disponible [Internet]: http://basededatos.umh.es/e_r.htm [Consultado Junio 2015]

[A.SILBERSCHATZ] Fundamentos de Bases de Datos (4ª edición). A. Silberschatz, H. F. Korth y S. Sudarshan. McGraw Hill, 2002

[D.CUADRA] Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación. Dolores Cuadra, Elena Castro, etc. RA-MA, 2007

[API_FACEBOOK] Documentación del API de Facebook para PHP. Disponible [Internet]: <https://developers.facebook.com/docs/reference/php/> [Consultado junio 2015]

[API_GOOGLE] Versión 3 del API de JavaScript de Google Maps. Cómo obtener una clave de API. Disponible [Internet]: https://developers.google.com/maps/documentation/javascript/tutorial?hl=es#api_key [Consultado Junio 2015]

Anexo I

Instalación

7.1 Instalación y configuración de una Aplicación en Facebook

Este capítulo trata de explicar los conceptos básicos para poder crear y configurar una aplicación en Facebook. Se explicará qué es la plataforma Facebook y cómo se relaciona con nuestra aplicación, y se hará un repaso de los elementos que proporciona la plataforma Facebook haciendo hincapié en las herramienta de testeo.

7.1.1 Plataforma Facebook

La plataforma Facebook proporciona una interfaz entre nuestra aplicación y Facebook y es esencial para poder crear una aplicación dentro de la red social.

El modelo típico de una aplicación Web es aquel en el cual nuestra aplicación y sus datos están alojados en un Servidor Web, y los clientes mediante de sus navegadores, acceden a la misma a través de Internet.

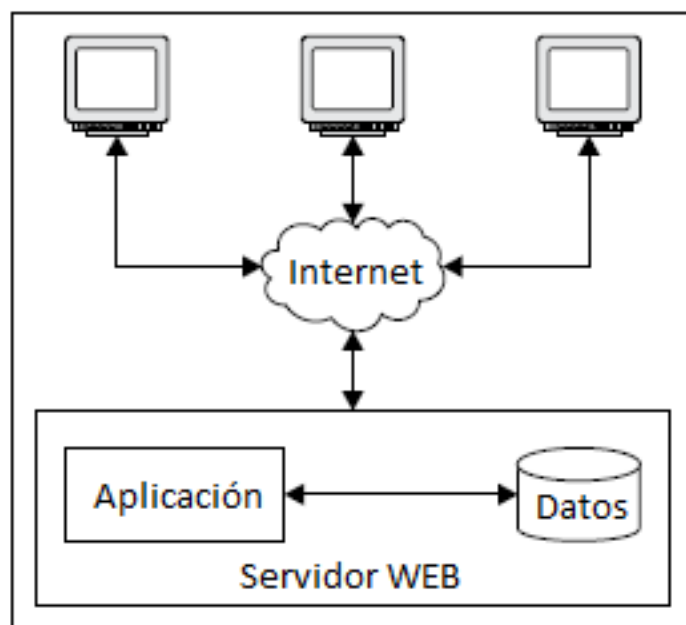


Ilustración 60 - Modelo típico de Aplicación Web

El modelo de Aplicación Web de Facebook es un poco diferente, ya que permite a los usuarios acceder a través de Facebook.com a nuestra aplicación, pero ésta nunca estará alojada en un servidor de Facebook, sino en nuestro servidor.

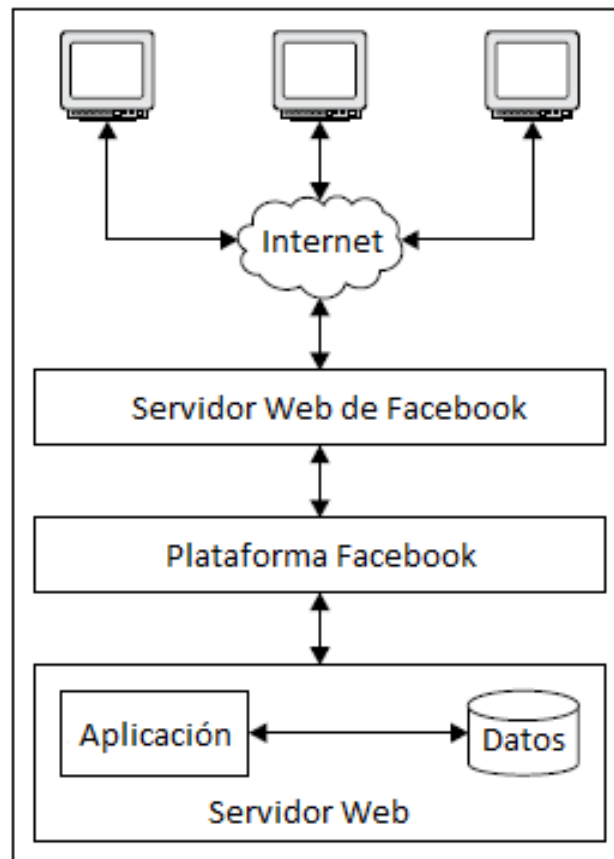


Ilustración 61 - Modelo de Aplicación Web de Facebook

La plataforma de Facebook está compuesta por dos elementos principales:

API de Facebook (API, del acrónimo en inglés Application Programming Interface, o Interfaz de Programación de Aplicaciones). Es un conjunto de librerías de software que permite trabajar con una aplicación (en este caso Facebook), sin tener que conocer nada sobre cómo funciona internamente.

FQL, o Facebook Query Language (Lenguaje de Consulta de Facebook). Los datos de Facebook se almacenan en Bases de Datos, por lo que se utiliza este lenguaje propio de Facebook para acceder a los mismos. Permite consultar tablas de la base de datos de Facebook directamente sin tener que recurrir a un método de la API.

Facebook proporciona además varias herramientas para el desarrollo de aplicaciones dentro de Facebook, de las cuales la más útil es Graph API Explorer, que sirve como introducción para conocer cómo funciona la API de Facebook. A continuación se detalla brevemente su funcionalidad.

7.1.2 Graph API Explorer

Esta herramienta permite realizar llamadas a través de la API de Facebook y sirve de interfaz gráfica para probar antes de desarrollar una aplicación los distintos métodos de la API y el formato de salida de las respuestas de la misma.

Para poder acceder a esta herramienta, es necesario tener una cuenta activa en Facebook, y haber accedido a la misma a través de la página de acceso de Facebook. Una vez se ha accedido a la cuenta en Facebook, la herramienta se encuentra en la siguiente URL: <http://developers.facebook.com/tools/explorer>

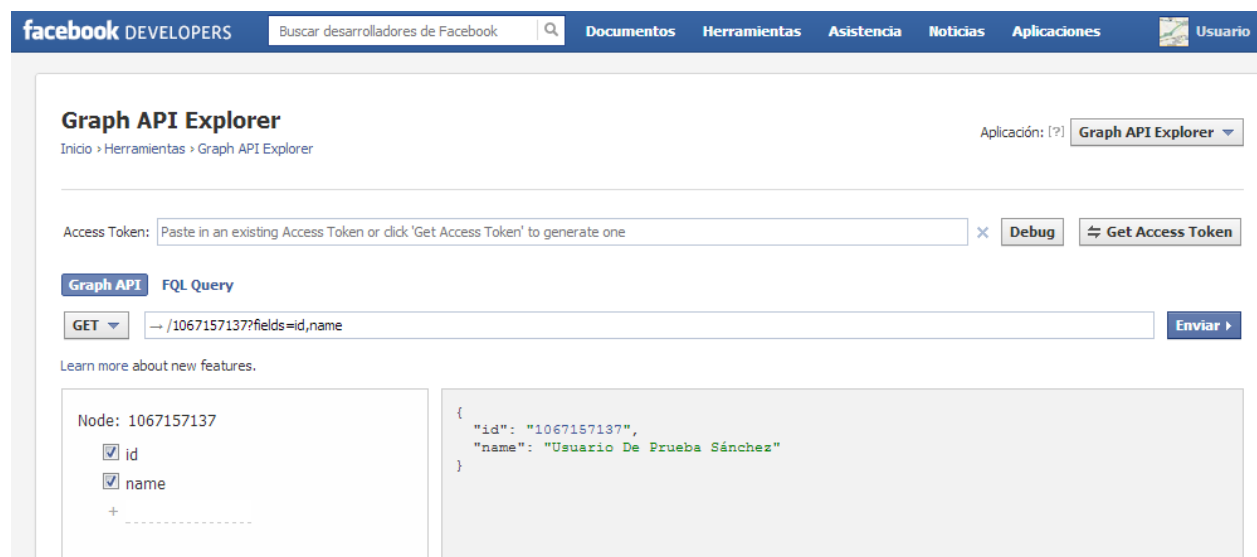


Ilustración 62 - Herramienta Graph API Explorer

Esta herramienta se compone de los siguientes campos:

Access Token (Token de Acceso): Es un código de respuesta que genera Facebook una vez el usuario se ha identificado y ha permitido el uso en una aplicación integrada en Facebook. Esta clave es obligatoria para la llamada de ciertos métodos de la API que necesitan permisos especiales y sin ella no será posible realizarlos.

Application: Aplicación a la cual está ligado el Token de Acceso que utilizará el API de Facebook para autenticar al usuario que realiza las llamadas a todos los métodos que necesiten de permisos específicos para su ejecución (obtener información personal, imágenes, amigos, post no públicos y un largo etcétera).

Botón Debug: Permite depurar un Token de Acceso para comprobar su validez, identificar al usuario válido, el origen del Token, etcétera.

Botón Get Access Token: Permite obtener un Token de Acceso con los permisos que se desee para una aplicación seleccionada.

Graph API: Permite realizar llamadas a métodos de la API. Contiene a su vez tres elementos:

Botón de selección de método de envío: Permite seleccionar el método de envío GET (para obtener datos), POST (para enviar datos) y DELETE (para eliminar)

Input de entrada de datos: Permite introducir el método y los parámetros de consulta que se realizará a través de la URL <https://graph.facebook.com> (todos los métodos disponibles y su formato están disponibles en la siguiente URL <http://developers.facebook.com/docs/reference/api/>).

Botón de Envío: Realiza el envío de los datos introducidos anteriormente.

FQL Query: Permite realizar consultas sobre la Base de Datos de Facebook con lenguaje FQL. Contiene dos elementos:

Input de entrada de datos: Permite introducir la sentencia FQL a ejecutar en el servidor de Base de Datos de Facebook. Se puede obtener un listado de todas las tablas disponibles y su definición en la URL <http://developers.facebook.com/docs/reference/fql/>

Botón de Envío: Realiza el envío de los datos introducidos anteriormente.

Salida: Muestra la salida de la consulta realizada.

Como ejemplo, en la siguiente imagen se muestra cómo hemos obtenido el identificador y el nombre de cada amigo en Facebook (en este caso sólo un amigo) mediante el método de la API `me/friends`. Previamente hemos obtenido un Token de Acceso válido para la aplicación Graph API Explorer de Facebook.

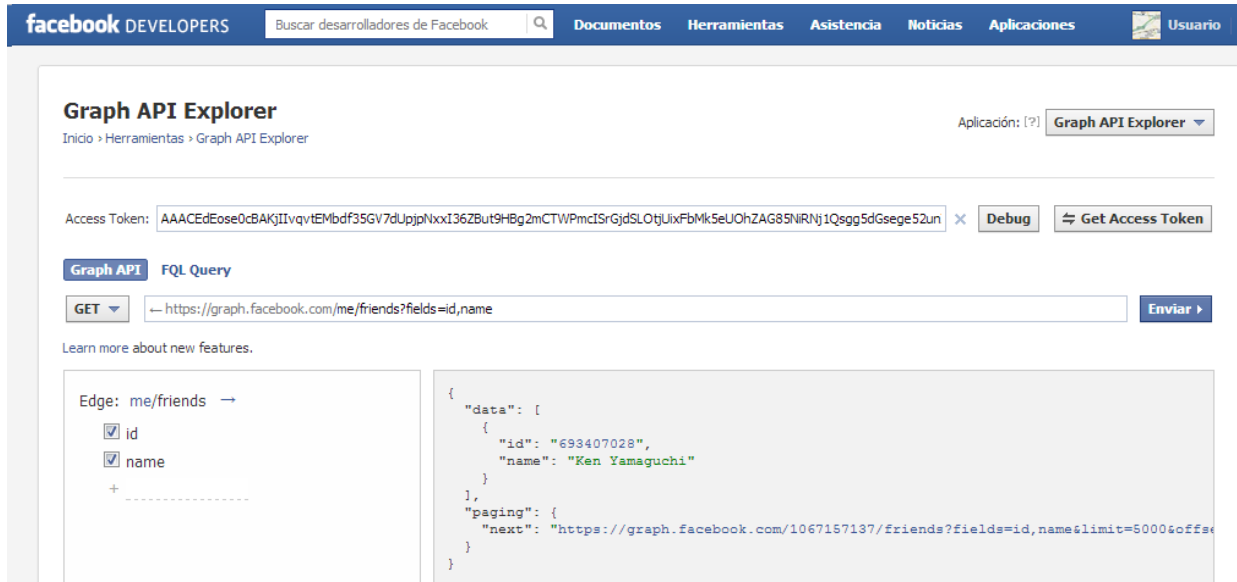


Ilustración 63 - Obtener datos de amigos mediante Graph API

En la siguiente imagen, hemos obtenido la misma información anterior utilizando FQL para obtener los datos directamente de la base de datos de Facebook. La sentencia que hemos utilizado es el siguiente:

```
select uid, name from user where uid in (
select uid2 from friend where uid1=me()
)
```

Previamente hemos obtenido un Token de Acceso válido para la aplicación Graph API Explorer.

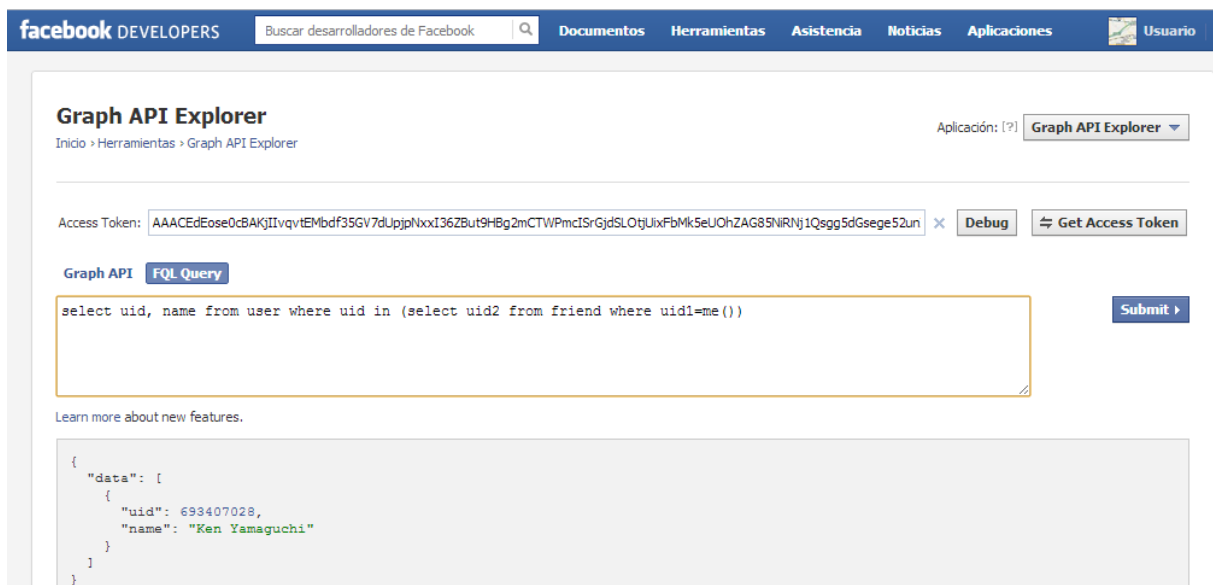


Ilustración 64 - Obtener datos de amigos mediante FQL**7.1.2.1 Obtener el Kit de Desarrollo de Facebook para PHP**

Facebook proporciona un Kit de Desarrollo en PHP con un conjunto completo de funcionalidades del lado del servidor para realizar llamadas a la API de Facebook, incluyendo todas las características de Graph API y FQL.

Además, mediante el uso de este Kit, se simplifica enormemente la autenticación y autorización de usuarios en nuestra aplicación, ya que se encarga de administrar los Token de Acceso de manera automática.

Para descargar el Kit de Desarrollo de Facebook es necesario acceder a la siguiente URL alojada en Github <https://github.com/facebook/facebook-php-sdk> y descargar la última versión disponible.

7.1.3 Instalar la aplicación Desarrollador

Para poder incluir aplicaciones en Facebook es necesario que previamente instalemos una aplicación llamada Desarrollador.

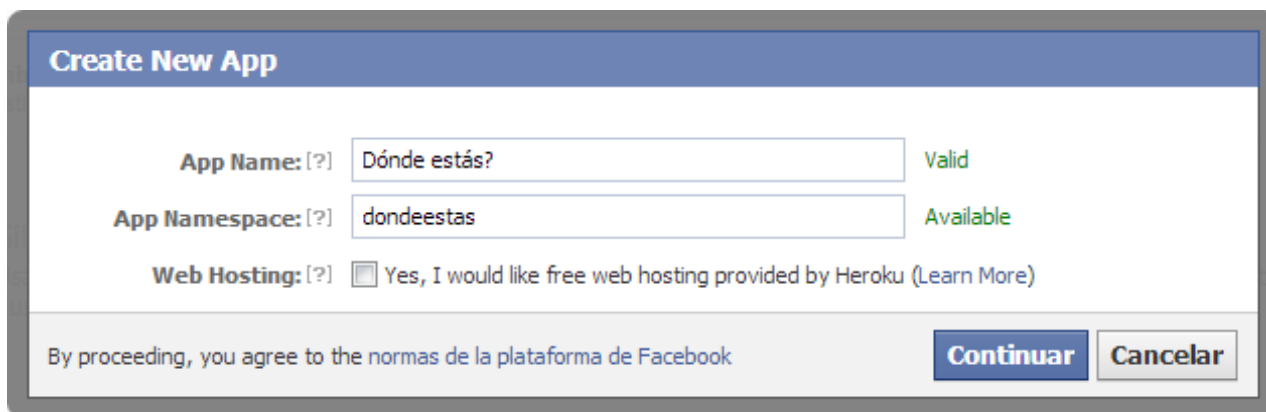
Esta aplicación permite añadir y configurar nuevas aplicaciones en Facebook. Para poder instalar la aplicación, se deberá acceder con una cuenta verificada a la URL <https://developers.facebook.com/apps> y registrarse como Desarrollador. Una vez registrado, se solicitará permiso para instalar la aplicación Desarrollador.



Ilustración 65 - Instalación de la aplicación Desarrollador

7.1.4 Configurar una nueva aplicación

Una vez instalada la aplicación Desarrollador, ya se puede comenzar a configurar nuevas aplicaciones. Para el desarrollo de este Proyecto de Fin de Carrera, se decidió crear una nueva aplicación en Facebook llamada “Dónde Estás?”, y utilizar el Namespace único “dondeestas”. El valor del campo Namespace se utiliza para definir acciones y objetos personalizados usando la API de Facebook, y para la URL personalizada de la aplicación en Facebook (<http://apps.facebook.com/dondeestas>).



Create New App

App Name: [?] Dónde estás? Valid

App Namespace: [?] dondeestas Available

Web Hosting: [?] ☒ Yes, I would like free web hosting provided by Heroku (Learn More)

By proceeding, you agree to the normas de la plataforma de Facebook

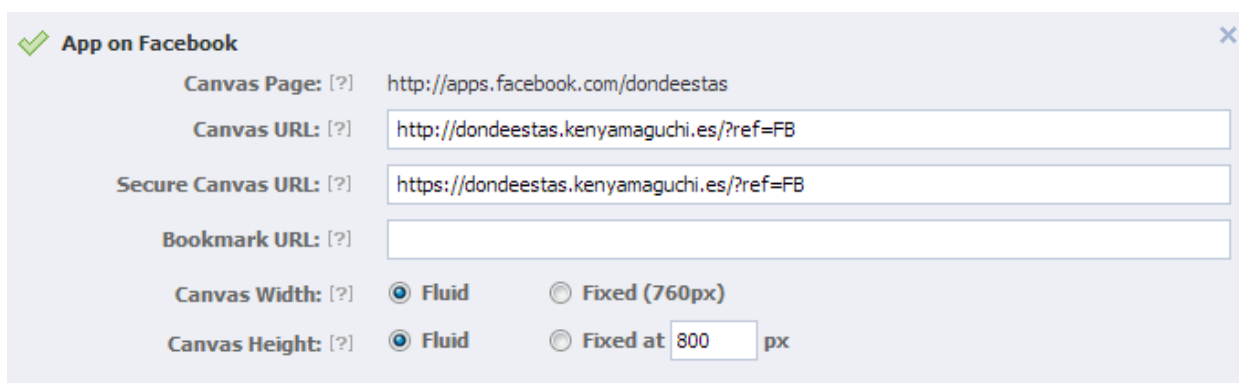
Continuar Cancelar

Ilustración 66 - Crear nueva aplicación

Una vez creada la nueva aplicación, es necesario configurar ciertos parámetros para que la aplicación funcione dentro de Facebook.

Las aplicaciones de Facebook se cargan dentro de Facebook en una sección llamada página de lienzo (Canvas Page) que básicamente es una página en blanco donde Facebook ejecuta nuestra aplicación. Cuando los usuarios hacen referencia a la página de lienzo de nuestra aplicación, Facebook la muestra conforme a su normativa.

Por lo tanto, uno de los parámetros que Facebook necesita para poder cargar nuestra aplicación, es la dirección URL de nuestra página de lienzo (y de nuestra página de lienzo segura). En este caso, se ha utilizado un servidor de pruebas accesible desde Internet con la siguiente URL <http://dondeestas.kenyamaguchi.es>



✓ App on Facebook

Canvas Page: [?] <http://apps.facebook.com/dondeestas>

Canvas URL: [?] <http://dondeestas.kenyamaguchi.es/?ref=FB>

Secure Canvas URL: [?] <https://dondeestas.kenyamaguchi.es/?ref=FB>

Bookmark URL: [?]

Canvas Width: [?] ☒ Fluid ☐ Fixed (760px)

Canvas Height: [?] ☒ Fluid ☐ Fixed at 800 px

Ilustración 67 - Configuración de la página de lienzo

Otra de los parámetros que serán necesarios para el correcto funcionamiento de la aplicación en Facebook son los permisos de la aplicación. Cuando nuestra aplicación solicita información a Facebook sobre el usuario que ejecuta la misma, puede necesitar de permisos adicionales para realizar ciertas acciones. Por ejemplo, nuestra aplicación puede publicar en el Muro del usuario una nueva posición en el mapa. Para ello, antes debemos

haber solicitado al usuario, mediante Facebook, un permiso concreto para poder realizar esta acción.

Los permisos que se necesitan configurar son los siguientes:

`user_about_me`: proporciona información básica del usuario (nombre, apellidos, imagen de usuario, URL del usuario...)

`friends_about_me`: proporciona información básica de los amigos del usuario (nombre, apellidos, imagen, URL...)

`publish_stream`: permite a la aplicación publicar contenido, comentarios y “me gusta” en el muro del usuario y de sus amigos. Este permiso entra dentro de la categoría de “Permisos Extendidos”, por lo que el usuario podrá decidir no permitir este tipo de acciones.

Configuring Permissions
Use these settings to enter the set of permissions your app requires when displayed in Centro de aplicaciones or using Auth Referrals
[Aprende más sobre Configuring Permissions](#)

Configure your permissions [Preview Auth Dialog](#)

Default Activity Privacy: [?] * Ninguna (opción predet... ▼

User & Friend Permissions: [?] user_about_me x friends_about_me x

Extended Permissions: [?] publish_stream x

Auth Token Parameter: [?] Query String (?code=...) ▼

Authenticated Referrals: [?] ☐ Activado ☒ Desactivado

Guardar cambios

Ilustración 68 - Permisos necesarios para la aplicación

Se puede consultar un listado de todos los permisos disponibles en la siguiente URL https://developers.facebook.com/docs/authentication/permissions/#user_friends_perms. Una vez hemos configurado los permisos en la aplicación, este sería el resultado que vería un usuario que instala por primera vez nuestra aplicación.

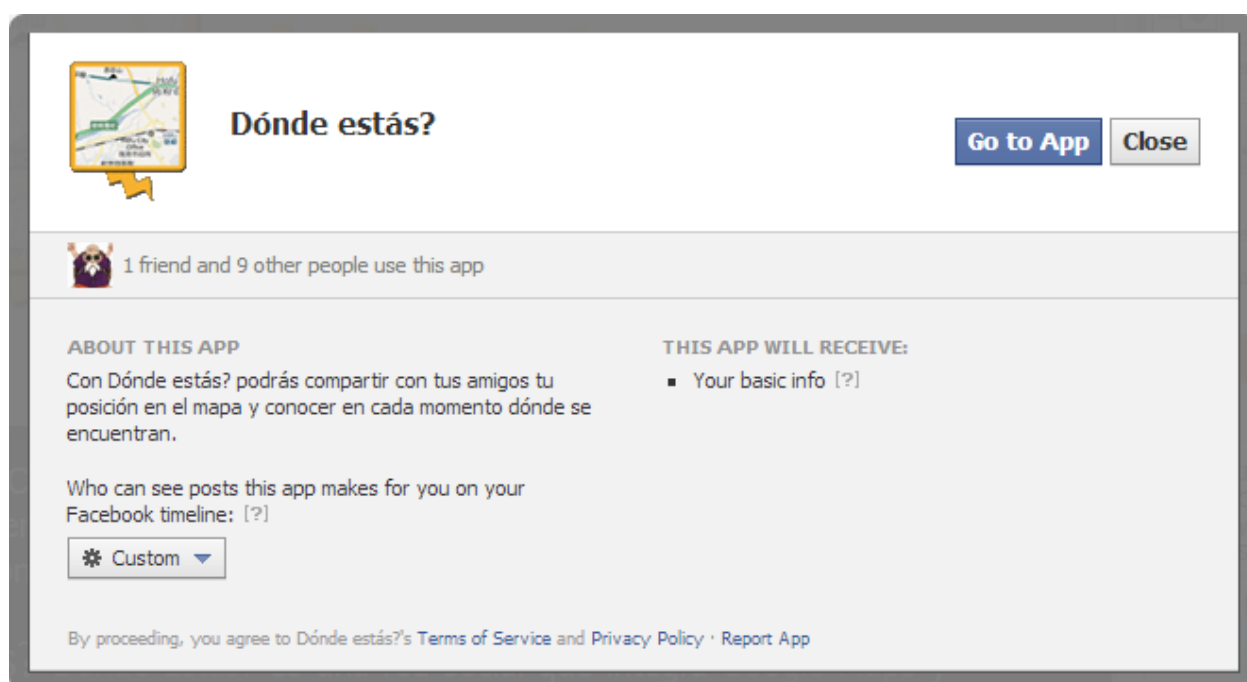


Ilustración 69 - Paso 1 autorización básica

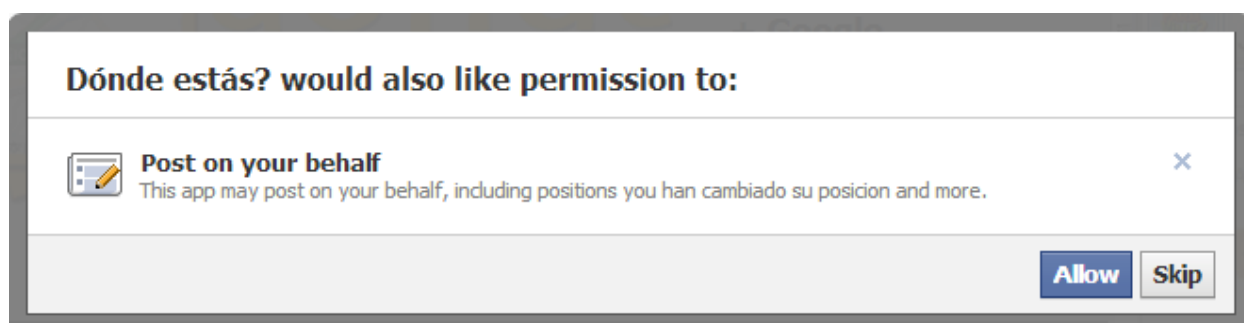


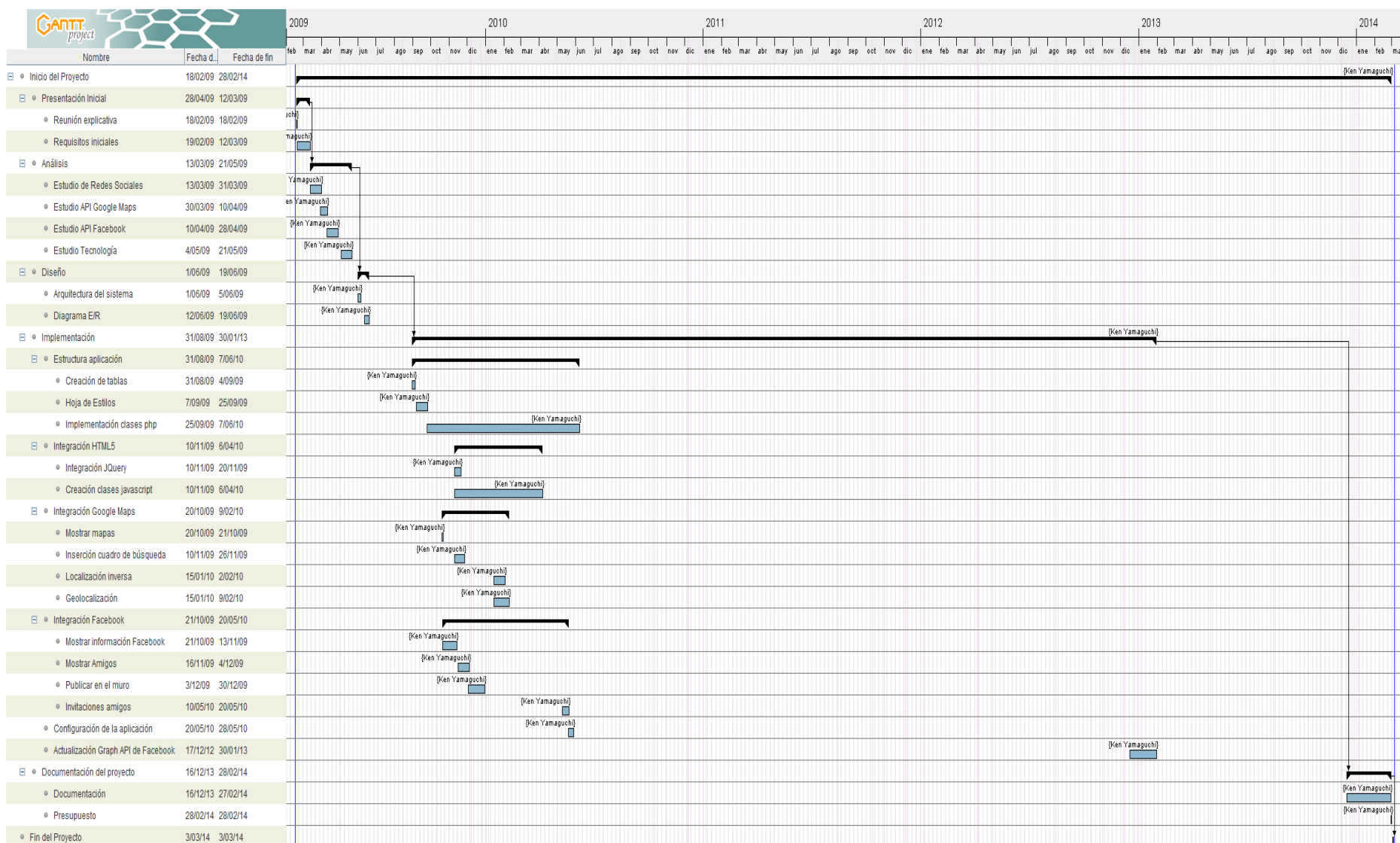
Ilustración 70 - Paso 2 autorización extendida

Anexo II

Diagrama Gantt

8.1 Diagrama Gantt

A continuación se muestra el diagrama Gantt utilizado para el proyecto.



ANEXO II